Digital Color Imaging

HANDBOOK

THE ELECTRICAL ENGINEERING AND APPLIED SIGNAL PROCESSING SERIES

Edited by Alexander Poularikas

The Advanced Signal Processing Handbook: Theory and Implementation for Radar, Sonar, and Medical Imaging Real-Time Systems
Stergios Stergiopoulos

The Transform and Data Compression Handbook
K.R. Rao and P.C. Yip

Handbook of Multisensor Data Fusion
David Hall and James Llinas

Handbook of Neural Network Signal Processing
Yu Hen Hu and Jenq-Neng Hwang

Handbook of Antennas in Wireless Communications
Lal Chand Godara

Noise Reduction in Speech Applications
Gillian M. Davis

Signal Processing Noise
Vyacheslav P. Tuzlukov

Digital Signal Processing with Examples in MATLAB®
Samuel Stearns

Applications in Time-Frequency Signal Processing
Antonia Papandreou-Suppappola

The Digital Color Imaging Handbook
Gaurav Sharma

Forthcoming Titles

Propagation Data Handbook for Wireless Communication System Design
Robert Crane

Smart Antennas
Lal Chand Godara

Pattern Recognition in Speech and Language Processing
Wu Chou and Bing Huang Juang

Nonlinear Signal and Image Processing: Theory, Methods, and Applications
Kenneth Barner and Gonzalo R. Arce

Forthcoming Titles *(continued)*

Soft Computing with MATLAB®
Ali Zilouchian

Signal and Image Processing Navigational Systems
Vyacheslav P. Tuzlukov

Wireless Internet: Technologies and Applications
Apostolis K. Salkintzis and Alexander Poularikas

Digital Color Imaging HANDBOOK

Edited by

Gaurav Sharma

Xerox Corporation
Webster, New York

CRC PRESS

Boca Raton London New York Washington, D.C.

Library of Congress Cataloging-in-Publication Data

Catalog record is available from the Library of Congress

Visit the CRC Press Web site at www.crcpress.com

International Standard Book Number 0-8493-0900-X
Printed in the United States of America 1 2 3 4 5 6 7 8 9 0
Printed on acid-free paper

Preface

The field of color imaging deals with the capture, processing, communication, and reproduction of color images. The origins of color imaging can be traced back to prehistoric times when cave dwellers created the first color drawings depicting events in their lives, using primitive materials and techniques available to them. Since then, color images have played an important role in history, and color imaging has advanced hand in hand with progress in science and technology. In the past 10 to 15 years, this field, like many others, has been significantly transformed by the digital revolution.

Digital color imaging devices such as digital still and video cameras, color scanners, displays, printers, DVD players, and cable/satellite set-top boxes are now commonplace in both home and office environments. A vast majority of color imagery is now captured digitally. An even larger fraction is digital during some part of the image's life cycle, so it is subject to computer-based processing. Digital technology enables unprecedented functionality and flexibility in the capture, processing, exchange, and output of color images. A knowledge of color science, color systems, appropriate processing algorithms, and device characteristics is necessary to fully harness this functionality and flexibility. As a result, the field of digital color imaging is a highly interdisciplinary area involving elements of physics, visual science, chemistry, psychophysics, computational algorithms, systems engineering, and mathematical optimization. While excellent texts and reference material exist in each of these areas, it has hitherto been the responsibility of researchers in the color imaging field to cull out relevant information. The goal of this handbook is to present aspects of these diverse elements as they relate to digital color imaging in a single and concise compilation. It is my hope that the handbook's assimilation of these different aspects and perspectives will aid students who are starting out in this area, as well as practitioners and researchers with expertise in specific domains who seek a better understanding of the rest of the system.

Chapters 1 through 3 are intended to cover the basics of color vision, perception, and physics that underpin digital color imaging. The material in these chapters will serve as useful background for those who are new to this area and as a refresher and update for color engineers with significant experience in the field. The end-to-end aspects of control and management of color in digital imaging systems are addressed in Chapter 4. Chapter 5 is

concerned with device color characterization, whereby the responses of individual color imaging devices (e.g., digital cameras, scanners, color printers, and displays) are measured and suitably accounted for in the capture and output of color images.

Chapters 6 and 7 address the important subject of digital halftoning, which deals with the rendition of images on printers and display devices that are capable of only bilevel reproduction or, more generally, of a limited number of levels. Since the vast majority of printers used in the printing and publishing industries are halftone printers, this topic is of significant interest in color imaging. Chapter 8 describes the compression of color images, which is a prerequisite for efficient use of network bandwidth and storage resources. The chapter cannot, and is not intended to, span the vast field of image compression. Instead, it focuses on aspects of image compression that are specifically pertinent to color images, a topic that is often left unaddressed by a number of image compression techniques. Brief overviews of the widely used JPEG and the emerging JPEG2000 image compression standards are included in the chapter.

Chapter 9 discusses color quantization or palettization of color images for use in frame-buffer systems with limited memory. While typical desktop displays today are "full-color" and typically do not require palettization, the issue is regaining importance in smaller displays on hand-held mobile devices, which are much more limited. Chapter 10 discusses techniques for pictorial gamut mapping. These techniques address the fundamental tradeoffs encountered when printing or displaying color images on common output devices that are capable of producing only a limited range of colors. Computationally efficient transforms for digital color imaging are discussed in Chapter 11. Finally, Chapter 12 covers color image processing in digital cameras, a topic that has assumed great importance with the explosion in the use of these devices for image capture.

Each chapter of the handbook is largely self-contained and can be read in isolation, provided the reader is generally familiar with the area. Cross-references among the chapters capture the important interrelationships in the information presented in the individual chapters. Chapter 1 also includes a broad overview of digital color imaging systems with references to, and connections between, the material in the other chapters, which may not be directly apparent. This is intended to facilitate the understanding of digital color imaging from a systems perspective, which is becoming increasingly important in today's open, interconnected world. Additional material related to the book will be made available on the publisher's web site www.crcpress.com. In particular, due to concerns of increased cost and the limitations of color accuracy in the printing process, a number of images that were originally in color have been included only as black-and-white figures in the book; full-color electronic versions of these figures are available online.

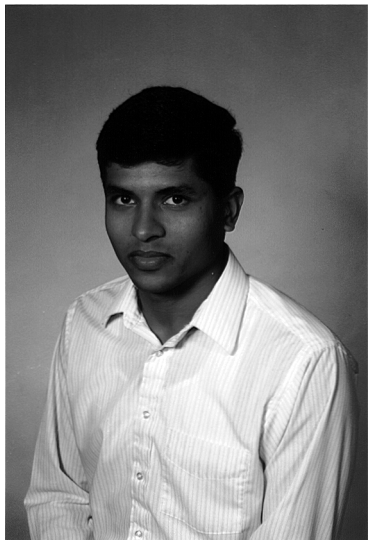
I would like to take this opportunity to thank all the authors for their excellent contributions. They have done an admirable job in writing for a

fairly wide audience while still communicating their individual research insights and accomplishments. The quality of the handbook can be directly attributed to their diligence.

I would also like to thank the outstanding staff at CRC press for their excellent support in the production and editing of this handbook. In particular, I would like to thank Nora Konopka for initiating this project, Helena Redshaw for urging me and the contributors to stay on schedule and for handling the submissions of all the materials, and Susan Fox for handling the copy editing and final production. Without their dedicated assistance, this project would have never been completed.

Gaurav Sharma
Xerox Corporation
Webster, NY
g.sharma@ieee.org

About the Editor



Gaurav Sharma is a member of the research staff at Xerox Corporation's Solutions and Services Technology Center, where he currently leads a research project on color imaging. He is also involved in teaching in an adjunct capacity at the Electrical and Computer Engineering Departments at the Rochester Institute of Technology, Rochester, New York. He received a BE degree in electronics and communication engineering from University of Roorkee, India, in 1990; an ME degree in electrical communication engineering from the Indian Institute of Science, Bangalore, India, in 1992; and an MS degree in applied mathematics and a Ph.D. degree in electrical and computer engineering from North Carolina State University, Raleigh, in 1995 and 1996, respectively.

From August 1992 through August 1996, he was a research assistant at the Center for Advanced Computing and Communications in the Electrical and Computer Engineering Department at North Carolina State University. His research and graduate work during this period focused on metrics for the evaluation and design of color recording devices. Since August 1996, he has been with Xerox Corporation. His research interests include color science and imaging, image security and halftoning, signal restoration, and error correction coding. Dr. Sharma is a member of Sigma Xi, Phi Kappa Phi, and Pi Mu Epsilon and is the current vice president of the Rochester chapter of the IEEE Signal Processing Society. He has authored or co-authored more than 40 technical papers in the fields of color, digital imaging, and image processing. He holds four U.S. patents and has more than a dozen pending U.S. patent applications.

Contributors

A. Ufuk Agar
Hewlett-Packard Laboratories
Color Imaging & Printing Technologies Department, HP Labs
Palo Alto, California

Jan P. Allebech
Purdue University
School of ECE
West Lafayette, Indiana

Raja Balasubramanian
Xerox Webster Research Center
Webster, New York

Farhan A. Baqai
Sony Corporation
Media Processing Division
San Jose, California

Luc Brun
Université de Reims Champagne Ardenne
Reims, France

Patrick Emmel
Clariant
Masterbatches Division
Muttenz, Switzerland

Mark D. Fairchild
Rochester Institute of Technology
Munsell Color Science Lab, Center for Imaging Science
Rochester, New York

Edward Giorgianni
Eastman Kodak Company
Imaging Research & Advanced Development Division
Rochester, New York

Charles Hains
Xerox Corporation
Webster, New York

Garrett M. Johnson
Rochester Institute of Technology
Center for Imaging Science
Rochester, New York

R. Victor Klassen
Xerox Corporation
Webster, New York

Keith Knox
Xerox Corporation
Xerox Digital Imaging Technology Center
Webster, New York

Thomas Madden
Eastman Kodak Company
Imaging Research & Advanced
Development Division
Rochester, New York

Jan Morovic
University of Derby
Colour & Imaging Institute
Kingsway, Derby, England

Ken Parulski
Eastman Kodak Company
Digital & Applied Imaging Division
Rochester, New York

Ricardo L. de Queiroz
Xerox Corporation
Corporate Research & Technology
Webster, New York

Gaurav Sharma
Xerox Corporation
Webster, New York

Kevin E. Spaulding
Eastman Kodak Company
Imaging Research & Advanced
Development Division
Rochester, New York

Alain Trémeau
Université Jean Monnet
de Saint-Etienne
Saint-Etienne, France

Shen-Ge Wang
Xerox Corporation
Webster, New York

Contents

chapter one

Color fundamentals for digital imaging

Gaurav Sharma
Xerox Corporation

Contents

1.1 Introduction

In our daily lives, color images surround us in print, television, computer displays, photographs, and movies. While these color images are taken for granted by a majority of readers and viewers, their production engages an entire industry of scientists, engineers, and practitioners. A knowledge of fundamental color principles is central to the work of this industry. The purpose of this chapter is to provide a concise introduction to some of these fundamentals of color science, colorimetry, color technology, and color systems. The presentation in the chapter is organized as a progressive introduction of principles from a logical rather than historical perspective. While suitable references and background material are included, the purpose is not to exhaustively document historical development of the principles or necessarily trace concepts to primary originators.

The perception of color is the result of interaction between a physical stimulus; receptors in the human eye that sense the stimulus; and the neural

system and the brain; which are responsible for communicating and interpreting the signals sensed by the eye. This clearly involves several physical, neural, and cognitive phenomena, which must be understood so as to comprehend color vision completely. While research continues in the integration of all these aspects of color, significant success has been achieved in understanding the physical and (to a lesser extent) neural phenomena involved in color sensation. The first part of this chapter attempts to summarize the current understanding in these areas with particular emphasis on the aspects that are of interest in color imaging applications.

The second part of the chapter is a brief overview of color recording and reproduction devices, their underlying physical principles, and color characteristics. Color measuring instrumentation, digital image recording devices such as scanners and digital color cameras, and color reproduction devices such as displays and printers are described. The spectral and color characteristics of images are also briefly discussed. The third part of the chapter describes the concepts of device-independent color and color management. The final section offers concluding remarks on the content covered elsewhere in the chapter.

Where appropriate, each section begins with a description of general principles and then briefly discusses their application in color imaging applications. Several of the topics covered here are discussed in significant detail in later chapters, but the material here provides a broad system-wide overview and indicates the connections and interrelations that may otherwise not be apparent.

1.2 *Physical stimuli for color*

The physical stimulus for color is electromagnetic radiation in the visible region of the spectrum, which is commonly referred to as *light*. In air or a vacuum, the visible region of the electromagnetic spectrum is typically specified by the wavelength region between λ_{min} = 360 nm and λ_{max} = 830 nm. Light stimulates retinal receptors in the eye, which ultimately causes the phenomenon of vision and the perception of color.

Our current understanding about the nature of light and color can be traced to the work of Sir Isaac Newton.[215] Newton's careful experiments[215,216] with sunlight and a prism helped dispel existing misconceptions and led to the realization that light can be decomposed into a spectrum of *monochromatic* components that cannot be further decomposed. Accordingly, light is characterized physically by its spectral composition. Typically, the characterization takes the form of a spectral power distribution (SPD), which characterizes light by the distribution of power (or energy per unit time) as a function of wavelength.†

† Note that the selection of wavelength rather than frequency or wave number for the specification of spectral power distribution of light is a rather arbitrary choice but has become a commonly accepted convention in the photometry, color measurement, and imaging communities.

Absolute spectral power distributions for light emitted or reflected off a surface are specified typically in radiometric units of Watts per steradian per square meter.[253,335] In practice, absolute SPDs are rarely (if ever) required for the purposes of color measurement and specification, and relative SPDs, where the scale/units are arbitrary, are commonly used. Figure 1.1 illustrates the relative SPDs of typical daylight, cool white fluorescent office lighting, and an incandescent lamp. The abscissa on the plot indicates the wavelength, and the ordinate indicates the relative density of light power. The mathematical interpretation of the spectral power distribution is as follows: if $l(\lambda)$ denotes the spectral power distribution, the power in an infinitesimal interval $d\lambda$ centered about λ_0 is given by $l(\lambda_0)d\lambda$.

Light incident on the eye may originate in different ways. When viewing self-luminous objects, the light directly originates from the object being viewed. More commonly, the object being viewed is illuminated by an external light source, such as daylight outdoors, or light from a lamp/overhead fixture indoors. In such situations, the SPD of light entering the eye is the product of the SPD of the light source and the spectral reflectance of the object. If the SPD of the illuminating source is given by $l(\lambda)$, and the spectral reflectance of the object is $r(\lambda)$, the SPD of the reflected light is given by the product $l(\lambda)r(\lambda)$. A similar relation is applicable to objects such a slides that are viewed in transmission, where the spectral reflectance is replaced by the spectral transmittance $t(\lambda)$. It is worth noting that the above mathematical relation is based on an idealized model of illuminant–object interaction that does not account for several geometry/surface effects such as the combina-

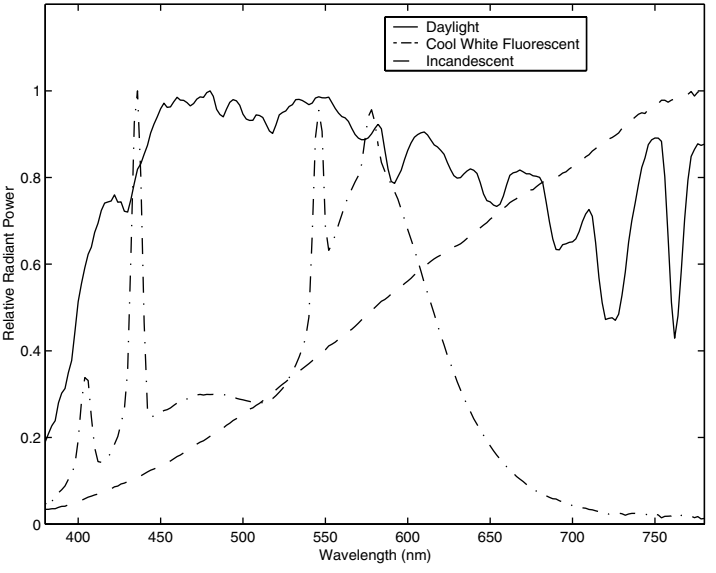


Figure 1.1 Measured relative spectral power distributions (SPDs) for daylight, cool white fluorescent office lighting, and an incandescent lamp.

tion of specular and body reflectance components.[189(pp. 43–45)] The model is, however, reasonably accurate for most imaging situations if care is taken to measure using a light source and geometry similar to that used in final viewing. Figure 1.2 illustrates a set of spectral reflectances for five different objects. One can see that the spectral reflectances of objects can demonstrate significant wavelength selectivity in that they reflect light of certain wavelengths with significantly more strength than light of other wavelengths. This spectral selectivity is typically the main determinant of the color appearance of the object.

1.2.1 The stimulus error

In discussing objects, it is common to say that they possess certain colors. For instance, the sky may be described as blue, an apple as red, and grass as green. In actuality, however, there is no color without an observer; therefore, attributing a color to an object is not strictly accurate. The attribution of colors to objects/lights is a particular instance of what psychologists refer to as the *stimulus error*[27,296] wherein a sensation experienced by an observer is identified with the stimulus causing the sensation. Color scientists and researchers have been aware of the stimulus error that pervades our common usage of color terms. Newton himself demonstrated this awareness in his quote, "The rays, to speak properly, are not colored; in them there is nothing else than a certain power and disposition to stir up a sensation of this or that color." Thus, speaking precisely, the light from the sky is not blue but evokes the sensation of blue when viewed by an observer.

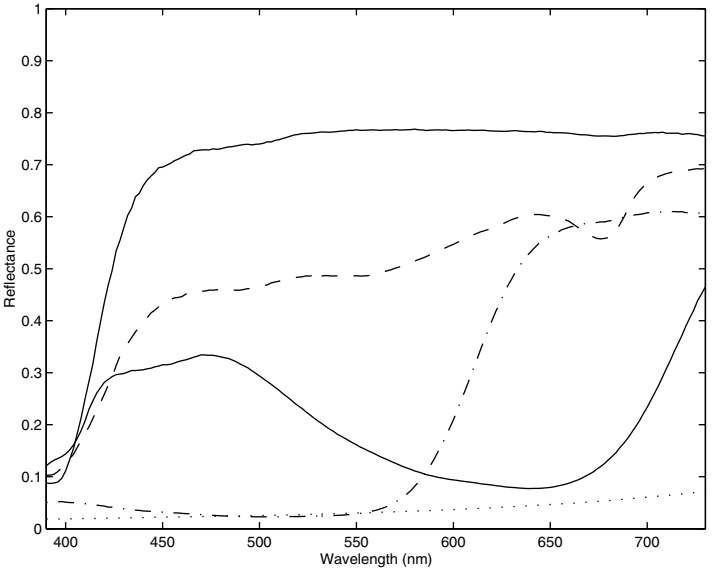


Figure 1.2 Measured spectral reflectance functions for five different natural objects.

As with several other sensations, the stimulus error is firmly ingrained in our usage of color terms, and one would have to go to great lengths and use awkward, pedantic language to avoid it entirely. Consequently, we continue to use color terms in commonly used contexts and talk, for instance, of cyan, magenta, and yellow colorants. It is, however, important to realize that an accurate interpretation of such statements requires a discounting of the stimulus error.

The stimulus error is often combined with other misuses of color terminology. For instance, one often hears the statement that a prism decomposes white light into its constituent colors. This statement is clearly inaccurate and unacceptable in technical usage. The proper statement would be that a prism decomposes light into its constituent spectral or wavelength components. Spectral power distributions of light, spectral reflectance functions, and spectral sensitivity functions are physical descriptions that are independent of observed sensation, and describing these in terms of color sensations is therefore incomplete and inaccurate. Errors of this type are therefore to be consciously avoided in technical descriptions of color.

1.3 *Human color perception and trichromacy*

Figure 1.3 shows a rough schematic of the human eye. The incident light is focused by the cornea and the eye's lens to form an image of the object being viewed onto the retina located at the back of the eyeball. The cornea provides most of the refraction needed to bring the light to a focus on the retina, and the primary purpose of the lens is to allow the eye to focus on objects at different viewing distances by changing the shape of the lens through the process of *accommodation*.[153(p. 100)] Photoreceptors within the retinal membrane are responsible for sensing the image and creating the neural signals that are responsible for the sense of sight. There are two kinds of photoreceptors: *rods* and *cones*. The rods are extremely sensitive to light and primarily useful for vision under very low light levels, termed as *scotopic* vision. In scotopic vision, only shades of gray can be perceived, and no color

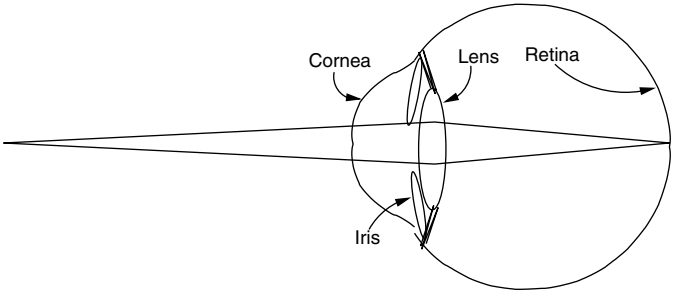


Figure 1.3 Schematic of the human eye.

is seen. This is the case, for instance, when objects are viewed under starlight. Under typical light levels used in imaging applications, the rods become saturated and do not contribute to vision; instead, the less-sensitive cones are active. The term *photopic* vision is used to describe this domain. There is a gradual change from photopic to scotopic vision as the illumination level is lowered, and in the intermediate *mesopic* form of vision both rods and cones are active. Typical light levels for these three domains of vision are listed in Section 1.5.1.

The cones are responsible for color vision. Observers with normal color vision† have three different types of cones, with photosensitive pigments that differ in their spectral absorption characteristics and, consequently, in their spectral sensitivities. The three types of cones are commonly called S, M, and L cones, which are abbreviated forms of short, medium, and long wavelength sensitive cones, respectively.‡ Under a fixed set of viewing conditions, the response of these cones can be accurately modeled by a linear system defined by the spectral sensitivities of the cones. If the spectral distribution of light incident on the retina is given by $f(\lambda)$, where λ represents wavelength (we are ignoring any spatial variations in the light for the time being), the responses of the three cones can be modeled as a three vector with components given by

$$c_i = \int_{\lambda_{min}}^{\lambda_{max}} s_i(\lambda) f(\lambda) d\lambda \qquad i = 1, 2, 3 \tag{1.1}$$

where $s_i(\lambda)$ denotes the sensitivity of the ith type of cones, and λ_{min}, λ_{max} denote the interval of wavelengths outside of which all these sensitivities are zero. As indicated earlier, in air or vacuum, this visible region of the electromagnetic spectrum is specified by the wavelength region between $\lambda_{min} = 360$ nm and $\lambda_{max} = 830$ nm. Estimates of the effective sensitivities of the LMS cones (i.e., cone fundamentals[256]) are shown in Figure 1.4.

Mathematically, the expressions in Equation 1.1 correspond to inner product operations[96] in the Hilbert space of square integrable functions $\mathcal{L}^2([\lambda_{min}, \lambda_{max}])$. Hence, the cone response mechanism corresponds to a projection of the spectrum onto the space spanned by three sensitivity functions $\{s_i(\lambda)\}_{i=1}^{3}$. This space is called the *human visual subspace* (HVSS).[55,56,125,304,310] The perception of color depends on further nonlinear processing of the retinal responses. However, to a first order of approximation, under similar conditions of adaptation, the sensation of color may be specified by the responses of the cones. This is the basis of all colorimetry and will be implicitly assumed throughout this section. A discussion of perceptual uniformity and appearance will be postponed until Sections 1.7 and 1.9.

† Around 8% of males and 0.5% of females are color deficient.

‡ Note that the common statement that the eye has three cones sensitive, respectively, to red, green, and blue light is not only inappropriate and erroneous for reasons described in Section 2.1, but also creates a circular definition.

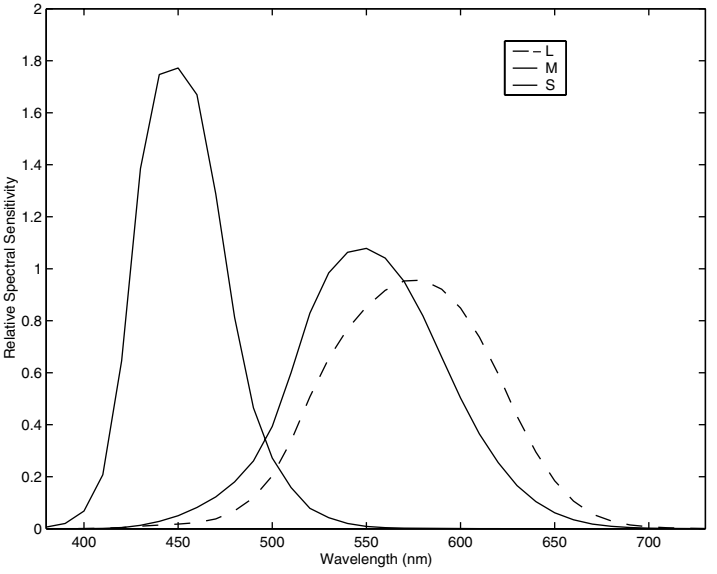


Figure 1.4 Estimated effective sensitivities of the L, M, S cones (cone fundamentals).

For computation, the spectral quantities in Equation 1.1 may be replaced by their sampled counterparts to obtain summations as numerical approximations to the integrals. For most color spectra, a sampling rate of 10 nm provides sufficient accuracy but, in applications involving fluorescent lamps with sharp spectral peaks, a higher sampling rate or alternative approaches may be required.[189,264,302,303] If N uniformly spaced samples are used over the visible range $[\lambda_{min}, \lambda_{max}]$, Equation 1.1 can be written as

$$c_i = \sum_{i=0}^{N-1} s_i(\lambda_i) f(\lambda_i) \Delta\lambda = \mathbf{s}_i^T \mathbf{f} \qquad i = 1, 2, 3 \tag{1.2}$$

In this equation, $\{\lambda_i\}_{i=0}^{N-1}$ are the uniformly spaced wavelengths covering the visible region of the spectrum, $\lambda_i = \lambda_0 + i\Delta\lambda$, with $\Delta\lambda$ as the wavelength sampling interval. The superscript T denotes the transpose operation, $\mathbf{f} = [f(\lambda_0), f(\lambda_1), \ldots, f(\lambda_{N-1})]^T$ is the $N \times 1$ vector of samples of $f(\lambda)$, and $\mathbf{s}_i = \Delta\lambda[s_i(\lambda_0), s_i(\lambda_1), \ldots, s_i(\lambda_{N-1})]^T$ is the $N \times 1$ vector of samples of $s_i(\lambda)$ scaled by the sampling interval $\Delta\lambda$. Note that, for notational simplicity, we have absorbed the influence of the sampling interval as a scaling factor into the cone sensitivity vectors $\{\mathbf{s}_i\}_{i=1}^3$. Equation 1.2 can be compactly written using matrix-vector notation as

$$\mathbf{c} = \mathbf{S}^T \mathbf{f} \tag{1.3}$$

where $\mathbf{c} = [c_1, c_2, c_3]^T$, $\mathbf{S} = [s_2, s_2, s_3]$ = the $N \times 3$ matrix with the cone sensitivity vectors as its columns. The HVSS then corresponds to the column space of $\mathbf{S}$.

In normal human observers, the spectral sensitivities of the three cones are linearly independent. Furthermore, the differences between the spectral sensitivities of color-normal observers are (relatively) small[277(p.343),328,335] and arise primarily due to the difference in the spectral transmittance of the eye's lens and the optical medium ahead of the retina.[71,211,219,220,328]

If a standardized set of cone responses is defined, color may be specified using the three-vector $\mathbf{c}$ in Equation 1.3, known as a *tristimulus* vector. Just as several different coordinate systems may be used for specifying position in three-dimensional space, any nonsingular, well-defined linear transformation of the tristimulus vector $\mathbf{c}$ can also serve the purpose of color specification. Because the cone responses are difficult to measure directly, but nonsingular linear transformations of the cone responses are readily determined through color-matching experiments, such a transformed coordinate system is used for the measurement and specification of color.

1.4 Color matching

Two spectra, represented by N-vectors $\mathbf{f}$ and $\mathbf{g}$, produce the same cone responses and therefore represent the same color if

$$\mathbf{S}^T\mathbf{f} = \mathbf{S}^T\mathbf{g} \tag{1.4}$$

Because $\mathbf{S}$ is an $N \times 3$ matrix with $N > 3$, the above system of equations has multiple solutions. This implies that many different spectra match in color.

It is, in fact, possible to draw significantly stronger conclusions from Equations 1.3 and 1.4. One of the characteristics of color vision that can be deduced based on these equations is the phenomenon of *trichromacy*, which states that it is possible to produce a color match for a given stimulus (equivalently, identical cone responses under the same viewing conditions) by using only combinations of light from three light sources.[105,200,201] To establish this, consider three color *primaries*, i.e., three *colorimetrically independent* light sources $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$. The term *colorimetrically independent* will be used in this chapter to denote a collection of spectra such that the color of any one cannot be visually matched by any linear combination of the others. Mathematically, colorimetric independence of $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ is equivalent to the linear independence of the three-vectors $\mathbf{S}^T\mathbf{p}_1$, $\mathbf{S}^T\mathbf{p}_2$, and $\mathbf{S}^T\mathbf{p}_2$. Hence, if $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3]$, the 3×3 matrix $\mathbf{S}^T\mathbf{P}$ is nonsingular.

For any visible spectrum $\mathbf{f}$ the three-vector

$$\mathbf{a}(\mathbf{f}) \stackrel{\text{def}}{=} (\mathbf{S}^T\mathbf{P})^{-1}\mathbf{S}^T\mathbf{f}$$

satisfies the relation

$$\mathbf{S}^T\mathbf{f} = \mathbf{S}^T\mathbf{P}\ \mathbf{a}(\mathbf{f}) \tag{1.5}$$

which is the relation for a color match. Hence, for any visible spectrum **f,** there exists a linear combination of the primaries, **P a**(**f**), which matches the color of **f**. This statement encapsulates the principle of trichromacy. It can be further seen that **a**(**f**) specifies the unique linear combination of primaries that matches **f** in color. This follows from the nonsingularity of $\mathbf{S}^T\mathbf{P}$, which ensures that if $\mathbf{S}^T\mathbf{f} = \mathbf{S}^T\mathbf{P}\mathbf{v}_1 = \mathbf{S}^T\mathbf{P}\mathbf{v}_2$, then $\mathbf{v}_1 = \mathbf{v}_2$. The elements of **a**(**f**) represent the relative intensities or "strengths" of the primaries required to match the color of **f**.

Some additional elaboration is necessary to establish the correspondence between the above mathematical argument and a physical experiment in which colors are matched using three primaries. In the mathematical computation, it is possible that the obtained vector of primary intensities, **a**(**f**), has negative components (in fact, it can be readily shown that, for any set of physical primaries, there exist visible spectra for which this happens). Because negative intensities of the primaries cannot be produced, the spectrum **P a**(**f**) is not realizable using the primaries. A physical realization corresponding to the equations is, however, still possible by rearranging the terms in Equation 1.5 and "subtracting" the primaries with negative intensities from **f**. The double negation cancels out and corresponds to the addition of positive amounts of the appropriate primaries to **f**.

The setup for a typical color-matching experiment is shown schematically in Figure 1.5. The observer views a small circular field that is split into two halves. The spectrum **f** is displayed on one half of a visual field. On the other half of the visual field appears a linear combination of the primary sources. The observer attempts to visually match the input spectrum by adjusting the relative intensities of the primary sources. The vector **a**(**f**) denotes the relative intensities of the three primaries when a match is obtained. Physically, it may be impossible to match the input spectrum by adjusting the intensities of the primaries. When this happens, the observer is allowed to move one or two of the primaries so that they illuminate the same field as input spectrum, **f** (see Figure 1.6). As noted earlier, this procedure is mathematically equivalent to subtracting that amount of primary from the primary field; i.e., the strengths in **a**(**f**) corresponding to the primaries that were moved are negative. As demonstrated in the last paragraph, all visible spectra can be matched using this method.

1.4.1 Color-matching functions

The linearity of color matching expressed in Equation 1.4 implies that, if the color tristimulus values for a basis set of spectra are known, the color values for all linear combinations of those spectra can be readily deduced. The unit intensity monochromatic spectra, given by $\{\mathbf{e}_i\}_{i=1}^{N}$, where $\mathbf{e}_i$ is an N-vector having a one in the ith position and zeros elsewhere, form a orthonormal basis in terms of which all spectra can be expressed. Hence, the color match-

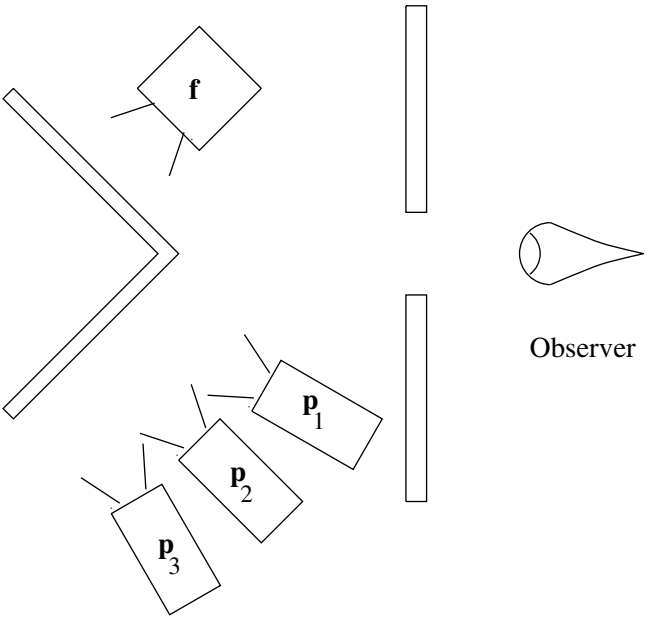


Figure 1.5 Color matching experiment.

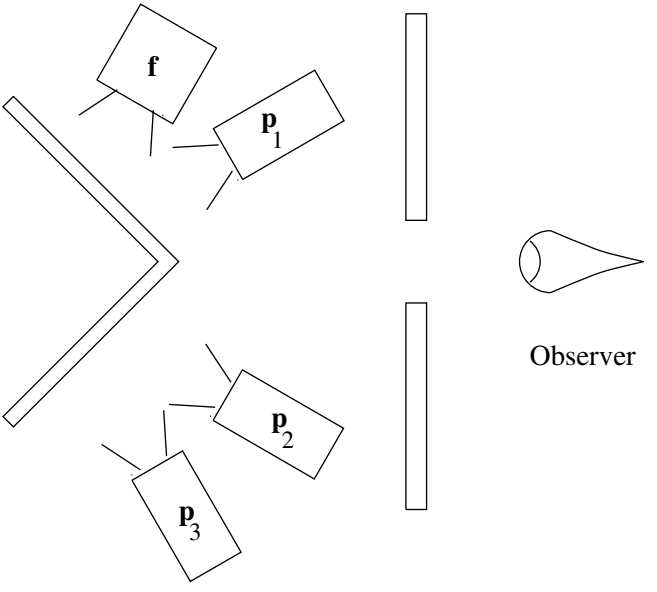


Figure 1.6 Color matching experiment with negative value for primary $\mathbf{p}_1$.

ing properties of all spectra (with respect to a given set of primaries) can be specified in terms of the color matching properties of these monochromatic spectra.

Consider the color matching experiment of the last section for the monochromatic spectra. Denoting the relative intensities of the three primaries required for matching $\mathbf{e}_i$ by $\mathbf{a}_i = \mathbf{a}(\mathbf{e}_i)$, the matches for all the monochromatic spectra can be written as

$$\mathbf{S}^T\mathbf{e}_i = \mathbf{S}^T\mathbf{P}\mathbf{a}_i \qquad i = 1, 2, \ldots, N \tag{1.6}$$

Combining the results of all N monochromatic spectra, we get

$$\mathbf{S}^T\mathbf{I}_N = \mathbf{S}^T\mathbf{P}\mathbf{A}^T \tag{1.7}$$

where $\mathbf{I}_N = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_N]$ is the $N \times N$ identity matrix, and $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_N]^T$ is the *color matching matrix* corresponding to the primaries $\mathbf{P}$.† The entries in the kth column of $\mathbf{A}$ correspond to the relative amount of the kth primary required to match $\{\mathbf{e}_i\}_{i=1}^N$, respectively. The columns of $\mathbf{A}$ are therefore referred to as the *color-matching functions* (CMFs) (associated with the primaries $\mathbf{P}$).

Now, reconsider the matching of a general spectrum $\mathbf{f} = [f_1, f_2, \ldots, f_N]^T$ in a color matching experiment using the primaries $\mathbf{P}$. The stimulus can be decomposed in terms of the unit intensity monochromatic stimuli $\{\mathbf{e}_i\}_{i=1}^N$ as

$$\mathbf{f} = \mathbf{I}_N\mathbf{f} = [\mathbf{e}_1\mathbf{e}_2\ldots\mathbf{e}_N][f_1, f_2, \ldots, f_N]^T = \sum_{i=1}^{N} f_i\mathbf{e}_i \tag{1.8}$$

Recall, a linear combination of the primaries with relative intensities specified by the tristimulus vector $\mathbf{a}_i$ matches the monochromatic spectrum $\mathbf{e}_i$. From the linearity of color matching and the above decomposition, it therefore follows that a linear combination of the primaries with relative intensities specified by the tristimulus vector

$$\sum_{i=1}^{N} f_i\mathbf{a}_i = \mathbf{A}^T\mathbf{f}$$

matches the spectrum $\mathbf{f}$. Thus, the tristimulus vector $\mathbf{A}^T\mathbf{f}$ represents the relative intensities of the primaries $\mathbf{P}$ that match the color of $\mathbf{f}$.

† In defining $\mathbf{A}$ as the matrix whose ith row is $\mathbf{a}_i^T$, we breach the common convention used throughout the rest of the chapter according to which a bold lower case subscripted letter denotes a column of the matrix denoted by the corresponding bold upper case letter.

From Equation 1.7, it can be readily seen that the color-matching matrix

$$\mathbf{A} = \mathbf{S}(\mathbf{P}^T\mathbf{S})^{-1} \tag{1.9}$$

Hence, the CMFs are a nonsingular linear transformation of the sensitivities of the three cones in the eye. It also follows that the color of two spectra, $\mathbf{f}$ and $\mathbf{g}$, matches if and only if $\mathbf{A}^T\mathbf{f} = \mathbf{A}^T\mathbf{g}$. As mentioned earlier, color of a visible spectrum, $\mathbf{f}$, may be specified in terms of the tristimulus values, $\mathbf{A}^T\mathbf{f}$, instead of $\mathbf{S}^T\mathbf{f}$. The fact that the color-matching matrix is readily determinable using the procedure outlined above makes such a scheme for specifying color considerably more attractive in comparison to one based on the actual cone sensitivities. Note also that the HVSS which was defined as the column space of $\mathbf{S}$ can alternately be defined as the column space of $\mathbf{A}$. Using Equation 1.9, we see that

$$\mathbf{A}^T\mathbf{P} = (\mathbf{S}(\mathbf{P}^T\mathbf{S})^{-1})^T\mathbf{P} = (\mathbf{S}^T\mathbf{P})^{-1}\mathbf{S}^T\mathbf{P} = \mathbf{I}_3 \tag{1.10}$$

where $\mathbf{I}_3$ is the 3×3 identity matrix. Equation 1.10 can also be obtained by direct reasoning. Consider a color matching experiment in which the stimulus to be matched by a combination of the primaries is one of the primaries itself, say $\mathbf{p}_1$. The unique values of the relative intensities of the primaries required to match $\mathbf{p}_1$ are $\mathbf{A}^T\mathbf{p}_1$. Because $\mathbf{p}_1 = \mathbf{P}[100]^T$ clearly matches itself, $\mathbf{A}^T\mathbf{p}_1 = [100]^T$. Similar relations hold for $\mathbf{p}_2$ and $\mathbf{p}_3$, and Equation 1.10 is obtained by concatenating the corresponding color match relations for all three primaries.

1.4.2 Metamerism and black space

As stated in Equation 1.4, two spectra represented by N-vectors $\mathbf{f}$ and $\mathbf{g}$ match in color if $\mathbf{S}^T\mathbf{f} = \mathbf{S}^T\mathbf{g}$ (or $\mathbf{A}^T\mathbf{f} = \mathbf{A}^T\mathbf{g}$). Because $\mathbf{S}$ (or equivalently $\mathbf{A}$) is an $N \times 3$ matrix, with $N > 3$, it is clear that several different spectra appear to be the same color to the observer. Two distinct spectra that appear the same are called *metamers*, and such a color match is said to be a *metameric match* (as opposed to a spectral match). Figure 1.7 shows plots of two metameric SPDs. Note that the colorimetry corresponding to these distributions is identical, but the SPDs exhibit very significant differences. The specific SPDs plotted here correspond to the SPD for CIE standard illuminant D65 (see Section 1.5.2) and a metameric match obtained to the corresponding SPD using typical CRT primaries.

The vector space view of color matching outlined above was first presented in a cohesive mathematical framework by Cohen and Kaupauf.[55–58] Tutorial descriptions using current notation and terminology appear in References 125, 299, 300, and 304. This approach allows us to deduce a number of interesting and useful properties of color vision. One such property is the decomposition of the N-dimensional spectral space into the three-dimensional HVSS and the $(N - 3)$-dimensional *metameric black space*, which was first hypothesized by Wyszecki.[332] Mathematically, this result states that any visible spectrum, $\mathbf{f}$, can be written as

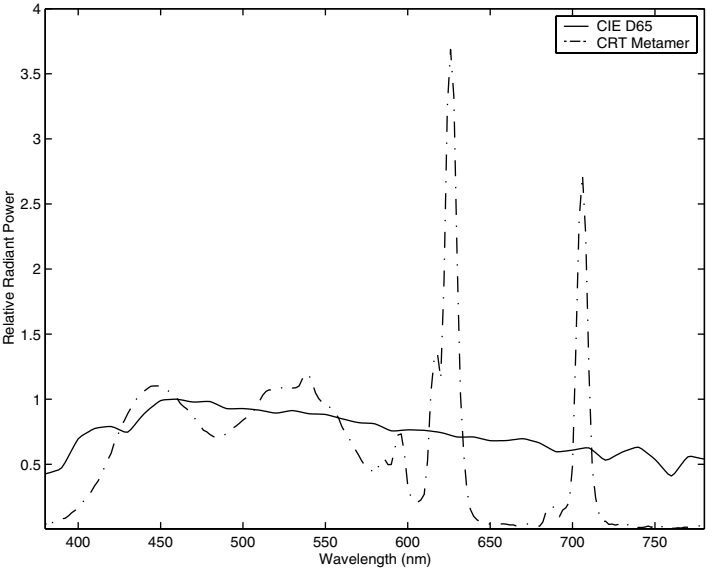


Figure 1.7 Example of a pair of metameric radiances.

$$\mathbf{f} = \mathbf{P}_\mathbf{A}\mathbf{f} + \mathbf{P}_\mathbf{A}^\perp\mathbf{f} \tag{1.11}$$

where $\mathbf{P}_\mathbf{A} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ is the orthogonal projector onto the column space of **A**, i.e., the HVSS, and

$$\mathbf{P}_\mathbf{A}^\perp = (\mathbf{I}_N - \mathbf{P}_\mathbf{A})$$

is the orthogonal projector onto the black space, which is the orthogonal complement of the HVSS. The projection, $\mathbf{P}_\mathbf{A}\mathbf{f}$, is called the *fundamental metamer* of **f,** because all metamers of **f** are given by

$$\left\{ \mathbf{P}_\mathbf{A}\mathbf{f} + \mathbf{P}_\mathbf{A}^\perp\mathbf{g} \quad | \quad \mathbf{g} \in R^N \right\}$$

Spectra that match in color have identical projections onto the HVSS. Conversely, spectra having identical projections onto the HVSS match in color. For a given spectrum **f**, the tristimulus value $\mathbf{t} = \mathbf{A}^T\mathbf{f}$ and the corresponding CMFs **A** can be used to compute the corresponding fundamental metamer as

$$\mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{t} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}(\mathbf{A}^T\mathbf{f}) = \mathbf{P}_\mathbf{A}\mathbf{f} \tag{1.12}$$

and, vice versa, the fundamental metamer $\mathbf{P_A f}$ and CMFs $\mathbf{A}$ can be used to compute the tristimulus,

$$\mathbf{t} = \mathbf{A}^T\mathbf{P_A f} = \mathbf{A}^T\mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{f} = \mathbf{A}^T\mathbf{f} \tag{1.13}$$

Equation 1.13 also illustrates the fact that the tristimulus values for a spectrum and its fundamental metamer are equivalent. Thus, the fundamental metamer offers an alternate representation of exactly the same information that is contained in the tristimulus values. The representation is, however, an N-vector in a three-dimensional subspace of the N-dimensional spectral space and therefore quite powerful and useful in the comparison of colors and spectra.[56] Tristimulus values are not ideally suited for the same task because of the dimensional mismatch between three-dimensional tristimulus values and N-dimensional spectra.

Another direct consequence of the above description of color matching is the fact that the primaries in any color matching experiment are unique only up to metamers. Because metamers are visually identical, the CMFs are not changed if each of the three primaries are replaced by any of their metamers.

The physical realization of metamers imposes additional constraints over and above those predicated by the equations above. In particular, any physically realizable spectrum needs to be non-negative, and hence it is possible that the metamers described by the above mathematics may not be realizable. In cases where a realizable metamer exists, set theoretic approaches may be used to incorporate non-negativity and other constraints.[261,299]

1.5 Colorimetry

It was mentioned in Section 1.4.1 that the color of a visible spectrum $\mathbf{f}$ can be specified in terms of the tristimulus values, $\mathbf{A}^T\mathbf{f}$, where $\mathbf{A}$ is a matrix of CMFs. To have agreement between different measurements, it is necessary to define a standard set of CMFs with respect to which the tristimulus values are stated. A number of different standards have been defined for a variety of applications, and it is worth reviewing some of these standards and the historical reasons behind their development.

1.5.1 CIE standards

The Commission Internationale de l'Eclairage (International Commission on Illumination, CIE) is the primary organization responsible for standardization of color metrics and terminology. A colorimetry standard was first defined by the CIE in 1931 and continues to form the basis of modern colorimetry. The CIE 1931 recommendations define a standard colorimetric observer by providing two different but equivalent set of CMFs. The first set of CMFs are known as the CIE RGB CMFs, $\bar{r}(\lambda)$, $\bar{g}(\lambda)$, $\bar{b}(\lambda)$. These are

associated with monochromatic primaries at wavelengths of 700.0, 546.1, and 435.8 nm, respectively, with their radiant intensities adjusted so that the tristimulus values of the equi-energy spectrum are all equal.[47] The equi-energy spectrum is the one whose SPD is constant (as a function of wavelength). The CIE RGB CMFs are shown in Figure 1.8.

The second set of CMFs, known as the CIE XYZ CMFs, $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, $\bar{z}(\lambda)$, are shown in Figure 1.9. They were recommended for reasons of more convenient application in colorimetry and are defined in terms of a linear transformation of the CIE RGB CMFs.[150] When these CMFs were first defined, calculations were typically performed on desk calculators, and the repetitive summing and differencing due to the negative lobes of the CIE RGB CMFs were prone to errors. Hence, the transformation from the CIE RGB CMFs to CIE XYZ CMFs was determined so as to avoid negative values at all wavelengths.[177] Because an infinite number of transformations can be defined to meet this non-negativity requirement, additional criteria were used in the choice of the CMFs.[85,153(p. 531)] Two of the important considerations were the choice of $\bar{y}(\lambda)$ coincident with the *luminous efficiency function*[335] and the normalization of the three CMFs so as to yield equal tristimulus values for the equi-energy spectrum. The luminous efficiency function gives the relative sensitivity of the eye to the energy at each wavelength. From the discussion of Section 1.4, it is readily seen that CMFs that are non-negative for all wavelengths cannot be obtained with any physically realizable primaries. Hence, any set of primaries corresponding to the CIE XYZ CMFs is not physically realizable. Table 1.1 provides a listing of the CIE XYZ color matching functions, sampled at 5-nm intervals in the range of 380 to 780 nm. Data used in this table are also available at the CIE web site.[47]

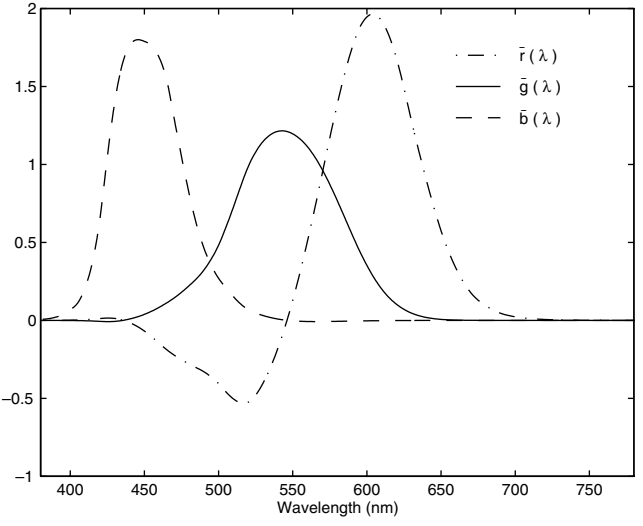


Figure 1.8 CIE $\bar{r}(\lambda), \bar{g}(\lambda), \bar{b}(\lambda)$ color matching functions.

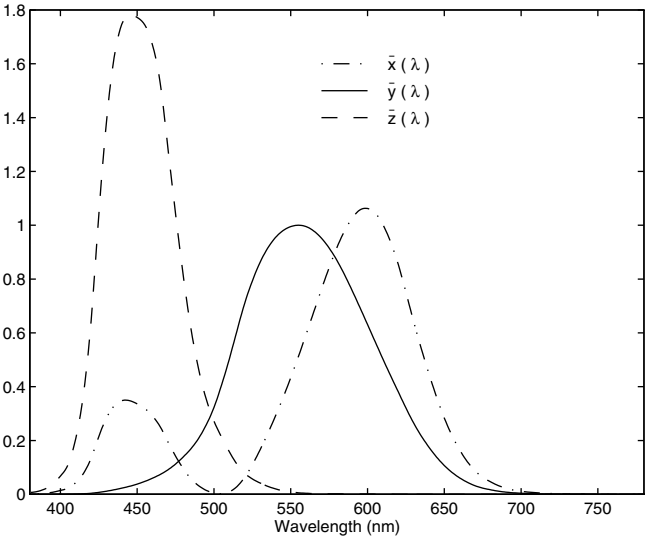


Figure 1.9 CIE $\bar{x}(\lambda), \bar{y}(\lambda), \bar{z}(\lambda)$ color matching functions.

The tristimulus values obtained with the CIE RGB CMFs are called the *CIE RGB tristimulus values*, and those obtained with the CIE XYZ CMFs are called the *CIE XYZ tristimulus values*. In most color imaging applications, and in color research, CIE XYZ values are used, and the CIE RGB tristimulus values are rarely used. The Y tristimulus value is usually called the *luminance* and correlates with the perceived brightness of the radiant spectrum. The luminance is described in units of candela per square meter (cd/m^2). Typical ambient luminance levels under sunlight, indoor lighting, moonlight, and starlight are of the order of 10^5, 10^2, 10^{-1}, and 10^{-3} cd/m^2, respectively. The scotopic, mesopic, and photopic domains of vision defined in Section 1.3 correspond roughly to luminance intervals 0.000001–0.034 cd/m^2, 0.034–3.4 cd/m^2, and over 3.4 cd/m^2, respectively.

The two sets of CMFs described above are suitable for describing color-matching when the angular subtense of the matching fields at the eye is between one and four degrees.[47,335(p. 131)] When the inadequacy of these CMFs for matching fields with larger angular subtense became apparent, the CIE defined an alternate standard colorimetric observer in 1964 with different sets of CMFs.[47] Because imaging applications (unlike quality control applications in manufacturing) involve complex visual fields where the color-homogeneous areas have small angular subtense, the CIE 1964 (10° observer) CMFs will not be discussed here.

1.5.2 Colorimetry for reflective objects

The discussion in the last section was based on the assumption that **f** is the spectral radiance of the light incident on the eye. Reflective objects are

Table 1.1 CIE XYZ CMFs for the 2° 1931 CIE Standard Observer and SPDs for Standard Illuminants D50, D65, and A Tabulated at 5-nm Wavelength Intervals (Except for the D50 SPD, these data are available from the CIE web site.[47])

Wavelength λ (nm)	$\bar{x}(\lambda)$	$\bar{y}(\lambda)$	$\bar{z}(\lambda)$	$I_{D50}(\lambda)$	$I_{D65}(\lambda)$	$I_A(\lambda)$
380	0.0014	0.0000	0.0065	24.4875	49.97550	9.795100
385	0.0022	0.0001	0.0105	27.1791	52.31180	10.899600
390	0.0042	0.0001	0.0201	29.8706	54.64820	12.085300
395	0.0076	0.0002	0.0362	39.5894	68.70150	13.354300
400	0.0143	0.0004	0.0679	49.3081	82.75490	14.708000
405	0.0232	0.0006	0.1102	52.9104	87.12040	16.148000
410	0.0435	0.0012	0.2074	56.5128	91.48600	17.675300
415	0.0776	0.0022	0.3713	58.2733	92.45890	19.290700
420	0.1344	0.0040	0.6456	60.0338	93.43180	20.995000
425	0.2148	0.0073	1.0391	58.9257	90.05700	22.788300
430	0.2839	0.0116	1.3856	57.8175	86.68230	24.670900
435	0.3285	0.0168	1.6230	66.3212	95.77360	26.642500
440	0.3483	0.0230	1.7471	74.8249	104.86500	28.702700
445	0.3481	0.0298	1.7826	81.0360	110.93600	30.850800
450	0.3362	0.0380	1.7721	87.2472	117.00800	33.085900
455	0.3187	0.0480	1.7441	88.9297	117.41000	35.406800
460	0.2908	0.0600	1.6692	90.6122	117.81200	37.812100
465	0.2511	0.0739	1.5281	90.9902	116.33600	40.300200
470	0.1954	0.0910	1.2876	91.3681	114.86100	42.869300
475	0.1421	0.1126	1.0419	93.2383	115.39200	45.517400
480	0.0956	0.1390	0.8130	95.1085	115.92300	48.242300
485	0.0580	0.1693	0.6162	93.5356	112.36700	51.041800
490	0.0320	0.2080	0.4652	91.9627	108.81100	53.913200
495	0.0147	0.2586	0.3533	93.8432	109.08200	56.853900
500	0.0049	0.3230	0.2720	95.7237	109.35400	59.861100
505	0.0024	0.4073	0.2123	96.1685	108.57800	62.932000
510	0.0093	0.5030	0.1582	96.6133	107.80200	66.063500
515	0.0291	0.6082	0.1117	96.8711	106.29600	69.252500
520	0.0633	0.7100	0.0782	97.1290	104.79000	72.495900
525	0.1096	0.7932	0.0573	99.6141	106.23900	75.790300
530	0.1655	0.8620	0.0422	102.0991	107.68900	79.132600
535	0.2257	0.9149	0.0298	101.4269	106.04700	82.519300
540	0.2904	0.9540	0.0203	100.7547	104.40500	85.947000
545	0.3597	0.9803	0.0134	101.5359	104.22500	89.412400
550	0.4334	0.9950	0.0087	102.3170	104.04600	92.912000
555	0.5121	1.0000	0.0057	101.1585	102.02300	96.442300
560	0.5945	0.9950	0.0039	100.0000	100.00000	100.000000
565	0.6784	0.9786	0.0027	98.8675	98.16710	103.582000
570	0.7621	0.9520	0.0021	97.7350	96.33420	107.184000
575	0.8425	0.9154	0.0018	98.3265	96.06110	110.803000
580	0.9163	0.8700	0.0017	98.9180	95.78800	114.436000

Table 1.1 CIE XYZ CMFs for the 2° 1931 CIE Standard Observer and SPDs for Standard Illuminants D50, D65, and A Tabulated at 5-nm Wavelength Intervals (Except for the D50 SPD, these data are available from the CIE web site.[47])

Wavelength λ (nm)	$\bar{x}(\lambda)$	$\bar{y}(\lambda)$	$\bar{z}(\lambda)$	$I_{D50}(\lambda)$	$I_{D65}(\lambda)$	$I_A(\lambda)$
585	0.9786	0.8163	0.0014	96.2084	92.23680	118.080000
590	1.0263	0.7570	0.0011	93.4988	88.68560	121.731000
595	1.0567	0.6949	0.0010	95.5933	89.34590	125.386000
600	1.0622	0.6310	0.0008	97.6878	90.00620	129.043000
605	1.0456	0.5668	0.0006	98.4784	89.80260	132.697000
610	1.0026	0.5030	0.0003	99.2691	89.59910	136.346000
615	0.9384	0.4412	0.0002	99.1553	88.64890	139.988000
620	0.8544	0.3810	0.0002	99.0415	87.69870	143.618000
625	0.7514	0.3210	0.0001	97.3817	85.49360	147.235000
630	0.6424	0.2650	0.0000	95.7218	83.28860	150.836000
635	0.5419	0.2170	0.0000	97.2895	83.49390	154.418000
640	0.4479	0.1750	0.0000	98.8572	83.69920	157.979000
645	0.3608	0.1382	0.0000	97.2622	81.86300	161.516000
650	0.2835	0.1070	0.0000	95.6672	80.02680	165.028000
655	0.2187	0.0816	0.0000	96.9285	80.12070	168.510000
660	0.1649	0.0610	0.0000	98.1898	80.21460	171.963000
665	0.1212	0.0446	0.0000	100.5966	81.24620	175.383000
670	0.0874	0.0320	0.0000	103.0034	82.27780	178.769000
675	0.0636	0.0232	0.0000	101.0682	80.28100	182.118000
680	0.0468	0.0170	0.0000	99.1330	78.28420	185.429000
685	0.0329	0.0119	0.0000	93.2570	74.00270	188.701000
690	0.0227	0.0082	0.0000	87.3809	69.72130	191.931000
695	0.0158	0.0057	0.0000	89.4922	70.66520	195.118000
700	0.0114	0.0041	0.0000	91.6035	71.60910	198.261000
705	0.0081	0.0029	0.0000	92.2460	72.97900	201.359000
710	0.0058	0.0021	0.0000	92.8886	74.34900	204.409000
715	0.0041	0.0015	0.0000	84.8715	67.97650	207.411000
720	0.0029	0.0010	0.0000	76.8544	61.60400	210.365000
725	0.0020	0.0007	0.0000	81.6828	65.74480	213.268000
730	0.0014	0.0005	0.0000	86.5112	69.88560	216.120000
735	0.0010	0.0004	0.0000	89.5455	72.48630	218.920000
740	0.0007	0.0002	0.0000	92.5798	75.08700	221.667000
745	0.0005	0.0002	0.0000	85.4048	69.33980	224.361000
750	0.0003	0.0001	0.0000	78.2299	63.59270	227.000000
755	0.0002	0.0001	0.0000	67.9609	55.00540	229.585000
760	0.0002	0.0001	0.0000	57.6918	46.41820	232.115000
765	0.0001	0.0000	0.0000	70.3074	56.61180	234.589000
770	0.0001	0.0000	0.0000	82.9230	66.80540	237.008000
775	0.0001	0.0000	0.0000	80.5985	65.09410	239.370000
780	0.0000	0.0000	0.0000	78.2740	63.38280	241.675000

viewed under an illuminating light source and, accordingly, their colorimetry is specified under a suitable illuminant. For the purposes of defining colorimetry, a reflective object can be represented by the N-vector, $\mathbf{r}$, of samples of its spectral reflectance $r(\lambda)$. When the object is viewed under an illuminant with SPD $l(\lambda)$, represented in sampled form by the N-vector $\mathbf{l}$, the resulting SPD at the eye is the product $l(\lambda)r(\lambda)$ of the illuminant SPD and the object reflectance, which can be represented in sampled form as the N-vector $\mathbf{Lr}$, where $\mathbf{L}$ is the diagonal illuminant matrix with entries from $\mathbf{l}$ along the diagonal. The CIE XYZ tristimulus values defining the color are therefore given by

$$\mathbf{t} = \mathbf{A}^T\mathbf{L}\mathbf{r} = \mathbf{A}_\mathbf{L}^T\mathbf{r} \tag{1.14}$$

where $\mathbf{A}$ is the matrix of CIE XYZ CMFs, and $\mathbf{A}_\mathbf{L} = \mathbf{LA}$. Color measurement for transmissive objects can be similarly defined in terms of their spectral transmittance. The color matching functions can be scaled by a common scale factor so that the Y stimulus value corresponds to the luminance in units of cd/m^2. However, as mentioned earlier, the absolute SPDs for the illuminant are rarely known or required in applications of colorimetry of reflective objects. In the colorimetry of reflective objects, it is therefore common to normalize the tristimulus values (or equivalently the CMFs) so that the Y coordinate is 100 for a *perfect reflector*, whose spectral reflectance is unity across all wavelengths. Because computation of CIE XYZ colorimetry is a basic step commonly employed in color imaging, it is useful to list this computation of CIE XYZ values explicitly:

$$\begin{aligned} X &= k\sum_{i=0}^{N-1} \bar{x}(\lambda_i)l(\lambda_i)r(\lambda_i) \\ Y &= k\sum_{i=0}^{N-1} \bar{y}(\lambda_i)l(\lambda_i)r(\lambda_i) \\ Z &= k\sum_{i=0}^{N-1} \bar{z}(\lambda_i)l(\lambda_i)r(\lambda_i) \end{aligned} \tag{1.15}$$

where $\{\lambda_i\}_{i=0}^{N-1}$ are the uniformly spaced wavelengths covering the visible region of the spectrum $\lambda_i = \lambda_0 + i\Delta\lambda$, with $\Delta\lambda$ as the wavelength sampling interval, and the normalization factor k given by

$$k = \frac{100}{\sum_{i=0}^{N-1} \bar{y}(\lambda_i)l(\lambda_i)} \tag{1.16}$$

In addition to the CMFs, the CIE has defined a number of standard illuminants for use in colorimetry of nonluminous reflecting objects. The relative SPDs of a number of these standard illuminants are shown in Figure 1.10. The corresponding values are also tabulated in Table 1.1. To represent different phases of daylight, a continuum of daylight illuminants has been defined[47] that are uniquely specified in terms of their *correlated color temperature* (CCT). Because the temperature of a blackbody radiator describes its complete spectral power distribution and thereby its color, it is commonly referred to as the color temperature of the blackbody. For an arbitrary illuminant, the CCT is defined as the color temperature of the blackbody radiator that is visually closest to the illuminant (in color).[335] The D65 and D50 illuminant spectra shown in Figure 1.10 are two daylight illuminants commonly used in colorimetry and have CCTs of 6500 and 5000 K, respectively. The CIE illuminant A represents a blackbody radiator at a temperature of 2856 K and closely approximates the spectra of incandescent lamps. Sources with lower CCT tend to be more red, whereas those with higher temperatures are bluer. Illuminants with similar CCT are assumed to be similar with regard to their color rendering of illuminated objects. This is, however, true only for illuminants whose spectra closely resemble that of a blackbody radiator, and other spectra that have identical CCT can have very different distributions and color rendering properties.[202] An example of the problem with the use of CCT for specifying the color-rendering properties of an illuminant is shown in Figure 1.11, where two synthesized illuminants are shown along with a reflectance spectrum measured from a cyan print sample. Though the illuminants have the same luminance and an identical CCT of 5000K, the color difference for the reflectance sample under the two illuminants is rather

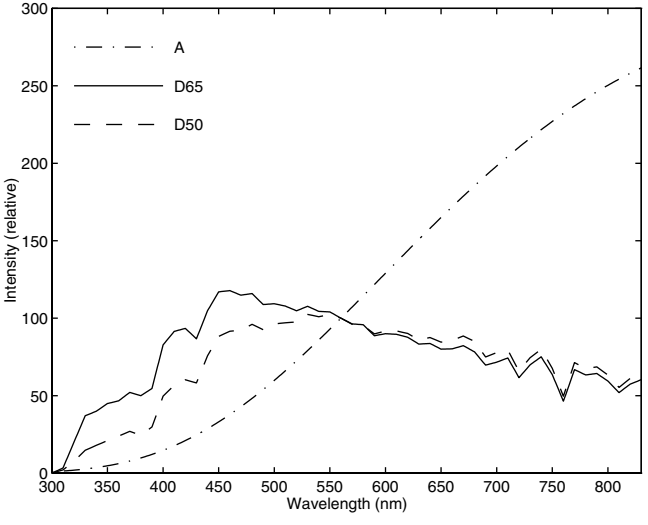


Figure 1.10 CIE standard illuminants.

large, corresponding to 44.4 ΔE^*_{ab} units. (For the definition of ΔE^*_{ab}, see Section 1.7.2.)

The definition of metameric matches and metamers can be extended to reflective objects. Two objects with (different) spectral reflectances $\mathbf{r}_1$ and $\mathbf{r}_2$ are said to be metamers (or in metameric match) under an illuminant with SPD $\mathbf{l}$ if

$$\mathbf{A}_\mathbf{L}^T\mathbf{r}_1 = \mathbf{A}_\mathbf{L}^T\mathbf{r}_2 \tag{1.17}$$

In analogy with the HVSS, the column space of $\mathbf{A}_\mathbf{L}$ is defined as the *Human Visual Illuminant Subspace* (HVISS).[310] In a fashion similar to that described in Section 1.4.2 for spectral radiances, the space of reflectances may also be decomposed into two orthogonal components, one being the HVISS and the other a black reflectance space, representing the absence of a visual stimulus. Every reflectance spectrum can then be represented as the summation of two orthogonal components, one in the three-dimensional HVISS and the other in the black reflectance space. Reflective metamers under a specified viewing illuminant have identical HVISS components, and their differences therefore lie entirely in the black reflectance space.

Metamerism is both a boon and a curse in color applications. Most color output systems (such as CRTs and color photography) exploit metamerism to reproduce color. However, in the matching of reflective materials, a metameric match under one viewing illuminant is usually insufficient to establish a match under other viewing illuminants. A common manifestation of this phenomenon is the color match of (different) fabrics under one illumination and mismatch under another. This situation is referred to as *illuminant metamerism*. Figure 1.12 shows an example of illuminant metamerism. The plots in this figure show the spectral reflectances of four different metameric samples that have identical colorimetry under CIE illuminant D50 but exhibit significant differences under other illuminants such as cool white fluorescent or CIE illuminant A. The four reflectances used in this example are spectral reflectances obtained with different color reproduction processes, representing one each of a photographic, xerographic, inkjet, and lithographic process. Details on how these metameric spectra were obtained can be found in Reference 270.

1.5.3 Chromaticity coordinates and chromaticity diagrams

Because color is specified by tristimuli, different colors may be visualized as vectors in three-dimensional space. However, such a visualization is difficult to reproduce on two-dimensional media and therefore inconvenient. A useful two-dimensional representation of colors is obtained if tristimuli are normalized to lie in the *unit plane*, i.e., the plane over which the tristimulus values sum up to unity. Such a normalization is convenient, as it destroys only information about the "intensity" of the stimulus and preserves com-

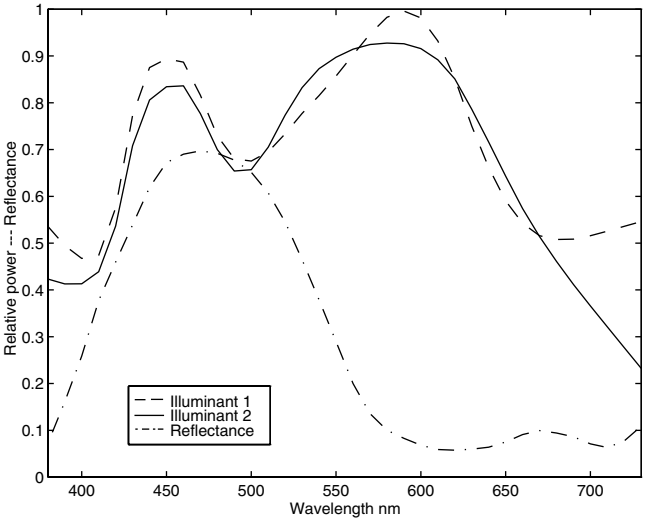


Figure 1.11 Correlated color temperature (CCT) counter-example with two illuminants with CCT = 5000 K, and a spectral reflectance.

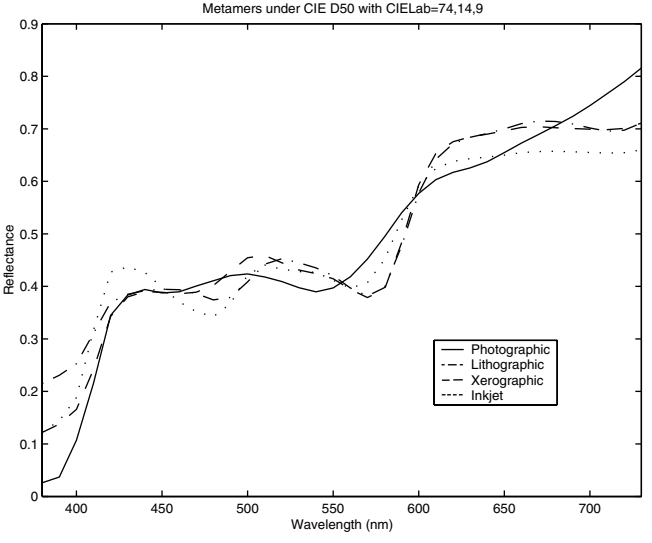


Figure 1.12 Reflective metamers under CIE illuminant D50 corresponding to different color reproduction processes.

plete information about the direction. The coordinates of the normalized tristimulus vector are called *chromaticity coordinates*, and a plot of colors on the unit plane using these coordinates is called a *chromaticity diagram*. Because the three chromaticity coordinates sum up to unity, typical diagrams plot only two chromaticity coordinates along mutually perpendicular axes.

The most commonly used chromaticity diagram is the CIE xy chromaticity diagram. The CIE xyz chromaticity coordinates can be obtained from the X,Y, Z tristimulus values in CIE XYZ space as

$$x = \frac{X}{X+Y+Z},$$

$$y = \frac{Y}{X+Y+Z},$$

$$z = \frac{Z}{X+Y+Z}, \tag{1.18}$$

Figure 1.13 shows a plot of the curve corresponding to visible monochromatic spectra on the CIE xy chromaticity diagram. This shark-fin-shaped curve, along which the wavelength (in nm) is indicated, is called the *spectrum locus*. From the linear relation between radiance spectra and the tristimulus values, it can readily be seen that the chromaticity coordinates of any additive-combination of two spectra lie on the line segment joining their chromaticity coordinates.[335] From this observation, it follows that the region of chromaticities of all realizable spectral stimuli is the convex hull of the spectrum locus. In Figure 1.13, this region of physically realizable chromaticities is the region inside the closed curve formed by the spectrum locus and the broken line joining its two extremes, which is known as the *purple line*.

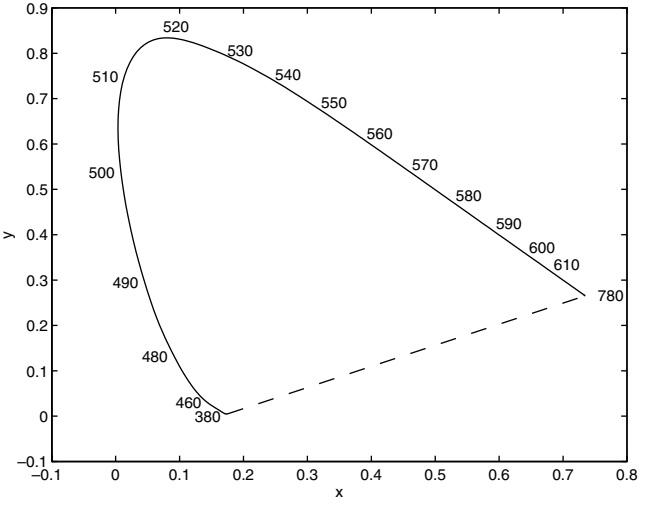


Figure 1.13 CIE xy chromaticity diagram.

1.5.4 *Transformation of primaries: NTSC, SMPTE, and CCIR primaries*

If a different set of primary sources, **Q**, is used in the color matching experiment, a different set of CMFs, **B**, is obtained. Because all CMFs are non-singular linear transformations of the human cone responses, the CMFs are related by a linear transformation. The relation between the two sets of CMFs can be obtained as follows. Using Equation 1.7 for the two sets of primaries and corresponding CMFs, both can be related to the eye's cone sensitivities and to each other as

$$\mathbf{S}^T = \mathbf{S}^T\mathbf{I}_N = \mathbf{S}^T\mathbf{P}\mathbf{A}^T = \mathbf{S}^T\mathbf{Q}\mathbf{B}^T \tag{1.19}$$

Post-multiplying both sides by $(\mathbf{S}^T\mathbf{P})^{-1}$ we have

$$(\mathbf{S}^T\mathbf{P})^{-1}\mathbf{S}^T = \mathbf{A}^T = (\mathbf{S}^T\mathbf{P})^{-1}\mathbf{S}^T\mathbf{Q}\mathbf{B}^T \tag{1.20}$$

Substituting the relation from the left side equality above into the right side yields

$$\mathbf{A}^T = \mathbf{A}^T\mathbf{Q}\mathbf{B}^T \tag{1.21}$$

Note that the columns of the 3×3 matrix $\mathbf{A}^T\mathbf{Q}$ are the tristimulus values of the primaries **Q** with respect to the primaries **P**. From the colorimetric independence of the primaries **Q**, it therefore follows that $\mathbf{A}^T\mathbf{Q}$ is non-singular, and we have

$$\mathbf{B}^T = (\mathbf{A}^T\mathbf{Q})^{-1}\mathbf{A}^T \tag{1.22}$$

Note that the same transformation, $(\mathbf{A}^T\mathbf{Q})^{-1}$, is useful for the conversion of tristimuli in the primary system ***P*** to tristimuli in the primary system ***Q***.

Color television was one of the first consumer products exploiting the phenomenon of trichromacy. The three light-emitting color phosphors in the television cathode ray tube (CRT) form the three primaries in this "color matching experiment." In the United States, the National Television Systems Committee (NTSC) recommendations for a receiver primary system based on three phosphor primaries were adopted by the Federal Communications Commission (FCC) in 1953 for use as a standard in color television (TV). The FCC standard specified the CIE xy chromaticity coordinates for the phosphors[68] as (0.670, 0.330) (red), (0.210, 0.710) (green), and (0.140, 0.080) (blue).[221] In addition, the tristimulus values (1, 1, 1) were assumed to correspond to a "white color" typically specified as the illuminant D65. The chromaticity coordinates along with the white balance condition define the CIE XYZ tristimuli of the NTSC primaries, which determine the relation of NTSC RGB tristimuli to CIE XYZ tristimuli as per Equation 1.22.

In the early color TV system, the signal-origination colorimetry was coupled with the colorimetry of displays, with the tacit assumption that processing at the receiver involves only decoding, and no color processing is performed. As display technology changed, manufacturers began using more efficient phosphors and incorporated some changes in the decoding as a compensation for the nonstandard phosphors.[225] Similar changes took place in the monitors used by broadcasters, but they were unaware of the compensating mechanisms in the consumer TV sets. As a result, there was considerable color variability in the broadcast TV system.[68] To overcome this problem, the chromaticities of a set of controlled phosphors was defined for use in broadcast monitors, and it now forms the Society of Motion Picture and Television Engineers (SMPTE) "C" phosphor specification.[279,280] Current commercial TV broadcasts in the U.S. are based on this specification.

With the development of newer display technologies that are not based on CRTs (see Section 1.11.1.5), it is now recognized that signal-origination colorimetry needs to be decoupled from the receiver colorimetry and that color correction at the receiver should compensate for the difference. However, for compatibility reasons and to minimize noise in transformations, it is still desirable to keep the reference primaries for broadcast colorimetry close to the phosphor primaries. Toward this end, the International Radio Consultative Committee (CCIR)[140] has defined a set of phosphor primaries by the chromaticity coordinates (0.640, 0.330) (red), (0.300, 0.600) (green), and (0.150, 0.060) (blue) for use in high-definition television (HDTV) systems.

Prior to transmission, tristimuli in SMPTE RGB and CCIR RGB spaces are nonlinearly compressed (by raising them to a power of 0.45) and encoded for reducing transmission bandwidth.[39,140] The reasons for these operations will be explained in Section 1.11.1.1. Note, however, that the encoding and nonlinear operations must be reversed before the signals can be converted to tristimuli spaces associated with other primaries. Transformations for the conversion of color tristimulus values between various systems can be found in References 234 (pp. 66–67), 142 (p. 71), and 231.

1.6 *Alternative color specification systems*

One of the limitations of the system of colorimetry outlined above is its nonintuitiveness and lack of clear relation to commonly understood color perception attributes such as hue, saturation, and lightness/brightness.† In describing perceived colors, most individuals resort to the use of color names such as white, black, red, green, yellow, blue, pink, etc. These terms, however, have no inherent ordering and are therefore limited in their utility unless they are conceptually organized into a *color order system*[330] based on perceptual principles.

† Readers are referred to Chapter 2 of this handbook or to References 132, 133, 135, and 335 (p. 487) for definitions of hue, chroma, saturation, lightness, brightness, and other color appearance terminology. Common notions of these terms will, however, suffice for the purposes of this chapter.

An early *color order system* for the specification of color based on the perceptual dimensions of hue, lightness, and chroma was developed by Munsell as a teaching aid for art students.[208] In the Munsell Color System, the hue, lightness, and chroma dimensions are described by the Munsell hue, Munsell value, and Munsell chroma, respectively.[25] A semi-numerical specification of the color is obtained by classifying colors into ten principal hues, with ten sub-hues for each hue, and a value between zero and ten. For defined hue and value, a chroma specification was experimentally obtained by selecting samples (with colors of the corresponding hue and value) of increasing chroma with equal perceived differences between neighboring samples. The step sizes for the perceptually equally spaced samples were determined so as to be consistent across different hue and value coordinates. A physical embodiment in the form of a color atlas[209] is an integral part of the Munsell system. The *Munsell Book of Color*, as the atlas was called, contains reflective samples that (when viewed under daylight) are spaced apart in perceptually equal steps of these attributes.[335] Colors in the Munsell system are specified by the combination of the Munsell hue, Munsell value, and Munsell chroma classifications/numbers. The Munsell system has undergone significant extension and evolution and is still in use.[209] In addition to the Munsell system, several other *color order systems* are in existence. The predominant among these are the Swedish Natural Color System[123,124,147,275,289] and the Optical Society of America (OSA) Uniform Color Scales (OSA-UCS) system.[1,186,187]

In the printing industry, it is common to create desired colors by using specially formulated colorants or premixed inks. These are typically known as *spot* colors. The colors are often communicated and specified by using printed samples that are organized by colorant and given distinct designations. Designers may thus choose a color from the available samples and communicate the color to printers using its designation, which specifies which ink is to be used in the printing process. The Pantone Matching System[224] is the main example of such a colorant-based empirical color specification system. Clearly, such a system has several limitations, the primary one being the variation in the specified "color" with a change in viewing illumination. Nonetheless, the system is in widespread use in the design and printing industries and has been extended to additional applications beyond printing.

The color specification systems described above are convenient for the specification of colors of uniform regions with reasonable spatial extent, such as those encountered in paints, color plastics, and textiles. The systems are therefore commonly used in the textiles and coloring industries. The color order systems are also commonly used in color research because of their desirable perceptual attributes. The Pantone Matching System is also commonly used for the specification of color in document imaging applications, typically for regions of uniform color such as a background or a corporate logo. These systems, however, they are not suited for the specification of colors in images where the colors are spatially and typically continuously

varying. A numerical system for color such as the CIE system is thus a requirement for imaging applications. The color order systems, however, play a significant role in the development and enhancement of the CIE numerical system; in particular, in the determination of color spaces with meaningful correlates of perceptual attributes and uniformity with respect to perception.

1.7 Uniform color spaces and color differences

The standards for colorimetry defined in Section 1.5 provide a system for specifying color in terms of tristimulus values that can be used to represent colors unambiguously in a three-dimensional space. It is natural to consider the relation of the distance between colors in this three-dimensional space to the perceived difference between them. Before such a comparison can be made, it is necessary to have some means for quantifying perceived color differences. For widely different color stimuli, an observer's assessment of the magnitude of color difference is rather variable and subjective.[335(p. 486)] At the same time, there is little practical value in quantifying large differences in color, and most research has concentrated on quantifying small color differences. For this purpose, the notion of a *just noticeably difference* (JND) in stimuli has been used extensively as a unit by color scientists.

Several researchers have examined the distribution of just noticeably different colors in CIE xy chromaticity and CIE XYZ tristimuli spaces and have found that it varies widely over the color space.[34,185,329,333,334] Hence, the CIE XYZ space is perceptually nonuniform in that equal perceptual differences between colors do not correspond to equal distances in the tristimulus space. Because perceptual uniformity is an extremely desirable feature for defining tolerances in color reproduction systems, considerable research has been directed toward the development of *uniform color spaces*. Traditionally, the problem has been decomposed into two sub-problems: one of determining a uniform lightness scale and the other of determining a uniform chromaticity diagram for equi-lightness color stimuli. The two are then combined with suitable scaling factors for the chromaticity scale and the lightness scale to make their units correspond to the same factor of a JND.

The historical milestones in the search for uniform brightness and lightness scales are described in Wyszecki and Stiles.[335(pp. 493–499)] Typical experiments determine these scales either by a process of repeated bisection of the scale extremes or by moving up in increments of a JND. Details of these standard psychophysical techniques can be found, for instance, in Gescheider.[100] A cube-root power law relation between brightness and luminance provides a satisfactory fit for most experimental data and therefore has the most widespread acceptance at present.[335(p. 494)]

The search for a uniform lightness scale was complemented by efforts toward determination of a *uniform chromaticity scale* for constant lightness. Two of these attempts are noteworthy. The first determined a linear transformation of the tristimulus space that yielded a chromaticity diagram with

just noticeably different colors being roughly equi-spaced.[151,152] This was the precursor of the CIE 1960 u,v diagram.[335(p. 503)] The second was primarily motivated by the Munsell system and used a nonlinear transformation of the CIE XYZ tristimuli to obtain a *chromatic-value* diagram in which the distances of Munsell colors of equal lightness would be in proportion to their hue and chroma differences.[4] The form for the nonlinear transformation was based on a color vision model proposed earlier by Adams,[3] and the diagram is therefore referred to as Adams' chromatic-value diagram.

Based on the aforementioned research, the CIE has recommended two uniform color spaces for practical applications: the CIE 1976 $L^*u^*v^*$ (CIELUV) space and the CIE 1976 $L^*a^*b^*$ (CIELAB) space.[47] These spaces are defined in terms of transformations from CIE XYZ tristimuli into these spaces. Both spaces employ a common lightness scale, L^*, that depends only on the luminance value Y. The lightness scale is combined with different uniform chromaticity diagrams to obtain a three-dimensional uniform color space. For the CIELUV space, a later version of the CIE 1960 u,v diagram is used, whereas CIELAB uses a modification of Adams' chromatic-value diagram.[335(p. 503)] In either case, the transformations include a normalization involving the tristimuli of a white stimulus, which provides a crude approximation to the eye's adaptation (see Section 1.9.1). Euclidean distances in either space provide a color-difference formula for evaluating color differences in perceptually relevant units. Both spaces also include correlates for the three perceptual attributes of lightness, chroma, and hue.

1.7.1 The CIE 1976 L*u*v* space

The L^*, u^*, v^* values corresponding to a stimulus with CIE XYZ tristimulus values X, Y, Z are given by[47]

$$L^* = 116 f\left(\frac{Y}{Y_n}\right) - 16 \tag{1.23}$$

$$u^* = 13L^*(u' - u'_n) \tag{1.24}$$

$$v^* = 13L^*(v' - v'_n) \tag{1.25}$$

where

$$f(x) = \begin{cases} x^{\frac{1}{3}} & x > 0.008856 \\ 7.787x + \dfrac{16}{116} & x \le 0.008856 \end{cases} \tag{1.26}$$

$$u' = \frac{4X}{X + 15Y + 3Z} \tag{1.27}$$

$$v' = \frac{9Y}{X + 15Y + 3Z} \tag{1.28}$$

$$u_n' = \frac{4X_n}{X_n + 15Y_n + 3Z_n} \tag{1.29}$$

$$v_n' = \frac{9Y_n}{X_n + 15Y_n + 3Z_n} \tag{1.30}$$

and X_n, Y_n, Z_n are the tristimuli of the white stimulus, which is typically the brightest stimulus in the field of view (see the next chapter for a more detailed discussion of adapting white, and also Section 1.9.1).

The Euclidean distance between two color stimuli in CIELUV space is denoted by ΔE^*_{uv} (delta E-uv) and is a measure of the total color difference between them. On average, a ΔE^*_{uv} value of around 2.9 corresponds to a JND.[192] As mentioned earlier, the value of L^* serves as a correlate of lightness. In the u^*v^* plane, the radial distance ($\sqrt{(u^*)^2 + (v^*)^2}$) and angular position ($\arctan\left(\frac{u^*}{v^*}\right)$) serve as correlates of chroma and hue, respectively.

*1.7.2 The CIE 1976 L*a*b* space*

The L^* coordinate of the CIELAB space is identical to the L^* coordinate for the CIELUV space, and the transformations for the a^* and b^* coordinates are given by

$$a^* = 500\left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right) \tag{1.31}$$

$$b^* = 200\left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right) \tag{1.32}$$

where $f(\cdot)$,X_n, Y_n, and Z_n are as defined earlier.

Because CIELAB is used extensively in imaging, transforms to and from CIELAB to other color spaces are commonly employed. For this reason, it is useful to list the inverse of the above transform, which converts a CIELAB value to a corresponding set of CIE XYZ values as follows:

$$f_Y = \frac{L^* + 16}{116} \tag{1.33}$$

$$f_X = \frac{a^*}{500} + f_Y \tag{1.34}$$

$$f_Z = f_Y - \frac{b^*}{200} \tag{1.35}$$

$$X = X_n \, f^{-1}(fx) \tag{1.36}$$

$$Y = Y_n \, f^{-1}(f_Y) \tag{1.37}$$

$$Z = Z_n \, f^{-1}(f_Z) \tag{1.38}$$

where f_Y, f_X, and f_Z are intermediate terms representing $f(Y/Y_n)$, $f(X/X_n)$, and $f(Z/Z_n)$, respectively, and $f^{-1}(\cdot)$ denotes the inverse of the function $f(\cdot)$ in Equation 1.26, given by

$$f^{-1}(t) = \begin{cases} t^3 & t > 0.206893 \\ \frac{1}{7.787}\left(t - \frac{16}{116}\right) & 0 \le t \le 0.206893 \end{cases} \tag{1.39}$$

In CIELAB, the radial distance in the a^*b^* plane

$$C^*_{ab} = \sqrt{(a^*)^2 + (b^*)^2} \tag{1.40}$$

serves as a correlate or measure of perceived chroma. The angular position in the a^*b^* plane

$$h^*_{ab} = \arctan\left(\frac{a^*}{b^*}\right) \tag{1.41}$$

serves as a correlate of perceived hue. Euclidean distance between two color stimuli in CIELAB space is denoted by ΔE^*_{ab} (delta E-ab). For a sample color with CIELAB values L^*_2, a^*_2, b^*_2, the color difference from a standard color with CIELAB values L^*_s, a^*_s, b^*_s is given by

$$\Delta E^*_{ab} = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2} \tag{1.42}$$

where $\Delta L^* = L^*_2 - L^*_s$, $\Delta a^* = a^*_2 - a^*_s$, and $\Delta b^* = b^*_2 - b^*_s$ are the distances of the sample from the standard along the L^*, a^*, and b^* axes, respectively. A ΔE^*_{ab} value of around 2.3 corresponds to a JND.[192] This correlation

is, however, quite approximate, and there are significant variations in a visual JND over color space.

The CIELAB space also incorporates an opponent-color encoding (see Sections 1.9 and 1.9.2 for details). The opponent encoding and the L^*, a^*, and b^* axes are illustrated in Figure 1.14. The a^* axis corresponds to red–green opponent hues, with distances along the positive a^* axis corresponding to a measure of redness and distances along the negative a^* axis corresponding to a measure of greenness. Likewise, the b^* axis corresponds to the yellow–blue opponent hues, with the distances along the positive b^* axis representing a measure of yellowness and distances along the negative b^* axis representing a measure of blueness. Thus, the CIELAB coordinates of a color can also provide a rough description for the color's perceptual attributes. A color with CIELAB of 50, 0, 0 is a mid-gray color with no apparent hue — commonly referred to as a *neutral* or *achromatic* color. A CIELAB value of 50, 0, 70 represents a mid-lightness saturated yellow color, and a CIELAB value of 90, –10, –7 represents a pastel cyan (bluish-green) color.

1.7.3 *Limitations of CIELAB and CIELUV spaces*

The CIELAB color space is widely used in color imaging and printing industries, and CIELUV is commonly used in the display industry. Both spaces, however, have several limitations. As may be expected, the CIELUV and CIELAB color spaces are only approximately uniform and are often inadequate for specific applications. The uniformity of CIELAB and CIELUV is about the same, but the largest departures from uniformity occur in different

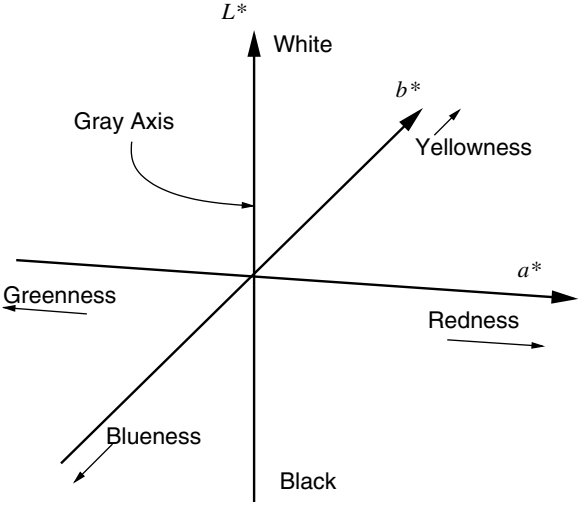


Figure 1.14 Interpretation of CIELAB axes and opponent nature of the a^* and b^* axes.

regions of the color space.[163,229,245] Several other uniform color spaces and color difference formulae have been proposed since acceptance of the CIE standards. Because CIELAB has gained wide acceptance as a standard, most of the difference formulae attempt to use alternate (non-Euclidean) *distance measures*† in the CIELAB space. Some of these are discussed in the following section.

The CIELAB and CIELUV spaces are also deficient with regard to the correlates for hue. Figure 1.15 shows a plot of the a^*–b^* plane where the loci of points corresponding to a constant perceived hue have been plotted. (The tables used to generate this figure were provided by Braun[33] and are based on visual data of Hung et al.[129]) The center of the figure corresponds to the L^* axis. Since the angular position

$$\arctan\left(\frac{a^*}{b^*}\right)$$

in CIELAB is a correlate of hue, one expects colors of a single hue should correspond to planes in CIELAB emanating radially outward from the CIELAB axis. The loci of points corresponding to a constant perceived hue in Figure 1.15 should therefore correspond to lines going radially outward from the center of the figure. This is, however, not the case. Particularly, the

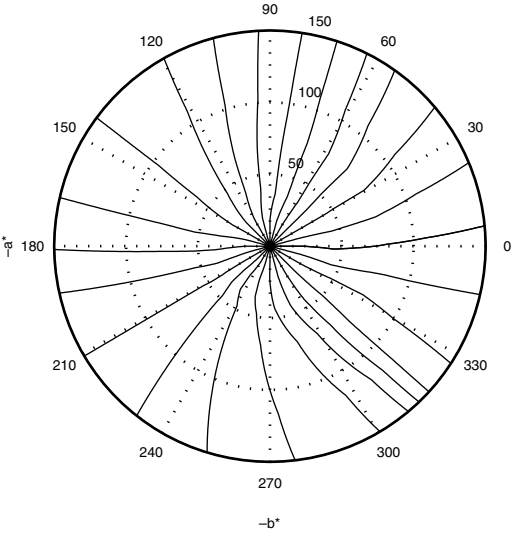


Figure 1.15 Loci of points corresponding to constant perceived hue in a^*-b^* plane (interpolated data).

† Note that several of these distance measures are asymmetric and as such do not satisfy the mathematical requirements for a metric.[96(p. 91)]

loci of points corresponding to constant perceived hues in the blue regions (in the vicinity of the negative b^* axis) show a very high curvature, indicating that the CIELAB hue correlate in this region is in poor agreement with actual perceived hue.

In several color processing operations, it is desirable to preserve a specified perceptual color attribute. A common example of this is the process of gamut-mapping (described in significant detail in a subsequent chapter), where it is often desirable to modify colors while preserving hue. Early attempts at preserving hue in the process of gamut mapping were based on the CIELAB space, where the gamut mapping was decomposed into individual mappings in CIELAB *hue planes*, i.e., planes emanating outward from the L^* axis. While these mappings preserve the CIELAB correlate of hue, the actual perceived hue is often poorly preserved due to the discrepancy between CIELAB hue and perceived hue described above. Understandably, the most significant artifacts are seen in the blue region, and the most common instantiation is the shift from blue to purple in the mapping of colors in the vicinity of the CRT blue primary to a printer gamut. More recently, the problem has been addressed by either warping the CIELAB space to correct for the hue "nonlinearity" before the gamut mapping[33] or through the use of alternate spaces, other than CIELAB, that provide better correlates of hue. Details are included in the chapter on gamut mapping.

1.7.4 Alternative color difference formulae

As indicated earlier, the Euclidean distance in CIELAB does not provide a very good agreement with the perceived magnitude of the color difference. Several alternative color difference formulae have therefore been proposed. Due to the widespread use of CIELAB, the most commonly used formulae are based on the CIELAB space and differ from ΔE^*_{ab} in that the computation of the distance between colors is not the Euclidean distance. Instead, the color difference formulae commonly decompose the Euclidean distance in CIELAB into components corresponding to differences in lightness, hue, and chroma, and these components are weighted differently to define a new color difference. Prominent among this class of uniform color difference formulae are the CMC(l:c) distance function,[52] the BFD(l:c) function,[182,183] the CIE-94 color difference formula,[49] and the recently developed CIE-DE2000 color difference formula.[179] The CMC(l:c) and the BFD(l:c) formulae have a long history and have influenced the development of the newer color difference formulae. These two color difference formulae are compared to ΔE^*_{ab} and a number of other uniform color spaces using perceptibility and acceptability criteria in Reference 192. In image processing applications involving color, the CIELAB and CIELUV spaces have been used extensively whereas, in industrial color control applications, the CMC formulae have found wider acceptance. In 1994, the CIE issued a new recommendation for the computation of color differences in CIELAB space that incorporates several of the robust and attractive features of the CMC(l:c) distance function.[49] Additional

features of the CMC and BFD color difference formulae have also been incorporated into CIE-DE2000.[179]

For the computation of the color difference of a sample color with CIELAB values L^*_2, a^*_2, b^*_2 from a standard color with CIELAB values L^*_s, a^*_s, b^*_s, a generic class of formula representing several of these color difference formulae can be written as[179]

$$\Delta E = \sqrt{\left(\frac{\Delta L^*}{k_L S_L}\right)^2 + \left(\frac{\Delta C^*}{k_C S_C}\right)^2 + \left(\frac{\Delta H^*}{k_H S_H}\right) + R_T \phi(\Delta C^* \Delta H^*)} \tag{1.43}$$

where k_L, k_C, k_H = positive, real-valued scaling parameters chosen based on the application where the formula is used

S_L, S_C, S_H = lightness-, chroma-, and hue-dependent scaling functions, respectively

R_T = an additional scaling function that depends on chroma and hue

The terms $\Delta L^*, \Delta C^*, \Delta H^*$ are referred to as the lightness, chroma, and hue differences, respectively. These are defined in terms of the standard and sample CIELAB values as†

$$\Delta L^* = L^*_2 - L^*_s \tag{1.44}$$

$$\Delta C^* = C^*_{ab,2} - C^*_{ab,s} \tag{1.45}$$

$$\begin{aligned} \Delta H^* &= \sqrt{(\Delta E^*_{ab})^2 - ((\Delta L^*)^2 + (\Delta C^*)^2)} \\ &= \sqrt{(\Delta a^*)^2 - (\Delta b^*)^2 + (\Delta C^*)^2} \\ &= \frac{2(a^*_2 b^*_s - a^*_s b^*_2)}{\sqrt{C^*_{ab,2} C^*_{ab,s} + a^*_2 a^*_s + b^*_2 b^*_s}} \end{aligned} \tag{1.46}$$

The sample and standard chroma values $C^*_{ab,2}$ and $C^*_{ab,s}$, respectively, are computed from the CIELAB coordinates as indicated in Equation 1.40, and the terms $\Delta L^*, \Delta a^*, \Delta b^*$, and ΔE^*_{ab} are as defined in Equation 1.42. The function $\phi(\cdot)$ is a function of the product of chroma and hue differences. Typically, $k_L = k_C = k_H = 1$ are used as the default parameter values in imaging applications.

For the CMC and CIE-94 color difference formulae, the R_T term in Equation 1.43 is absent, simplifying these formulae to

† The CIE-DE2000 formula requires an additional chroma-dependent scaling of the a^* axis prior to computation of the lightness, chroma, and hue differences.

$$\Delta E = \sqrt{\left(\frac{\Delta L^*}{k_L S_L}\right)^2 + \left(\frac{\Delta C^*}{k_C S_C}\right)^2 + \left(\frac{\Delta H^*}{k_H S_H}\right)^2} \tag{1.47}$$

The weighting functions S_L, S_C, and S_H for the CMC formula are computed from the CIELAB coordinates of the standard as[2,52,205]

$$S_L^{\text{CMC}} = \begin{cases} \dfrac{0.040975 L^*_s}{1 + 0.01765 L^*_s} & L^* \geq 16 \\ 0.511 & L^* < 16 \end{cases} \tag{1.48}$$

$$S_C^{\text{CMC}} = \frac{0.0638 C^*_{ab,s}}{1 + 0.0131 C^*_{ab,s}} + 0.638 \tag{1.49}$$

$$S_H^{CMC} = S_C^{CMC}(T^{CMC} F^{CMC} + 1 - F^{CMC}) \tag{1.50}$$

$$F^{CMC} = \sqrt{\frac{(C^*_{ab,s})^4}{(C^*_{ab,s})^4 + 1900}} \tag{1.51}$$

$$T^{CMC} = \begin{cases} 0.56 + |0.2 \cos(h^*_{ab,s} + 168°)| & 164° \leq h^*_{ab,s} \leq 345° \\ 0.36 + |0.4 \cos(h^*_{ab,s} + 35°)| & \text{otherwise} \end{cases} \tag{1.52}$$

where the CIELAB hue angle of the standard, $h^*_{ab,s}$, is computed in the interval [0, 360°] according to the definition in Equation 1.41. The parameter k_H is unity for the CMC color difference formula, and the parameters k_L and k_C (defined by the user) weight the importance of lightness and chroma relative to hue. The resulting color difference computed using the above scaling functions in Equation 1.47 is referred to as the $\Delta E_{CMC(k_L, k_C)}$ color difference. Note that the definitions of the lightness, hue, and chroma weighting functions in terms of the CIELAB coordinates of the standard make the CMC color formula asymmetric in that the distance between two samples is dependent on which one is chosen as the standard and which one as the sample.

The complexity of the CMC formula makes it difficult to understand intuitively based on equations alone. A graphical illustration of the CMC difference is therefore presented in Figure 1.16. The figure represents the CIELAB a^*–b^* plane, with the a^* and b^* axes as shown. The figure includes plots of several elliptical closed curves that represent the loci of points whose color difference with respect to a standard color at the "center" of the "ellipse" is one $\Delta E_{CMC(1,1)}$ unit. The "center" representing the standard has been excluded from the figure for clarity. Note that the figure is representative of a^*–b^* planes at all values of L^*, because the S_H^{CMC} and S_C^{CMC} weighting

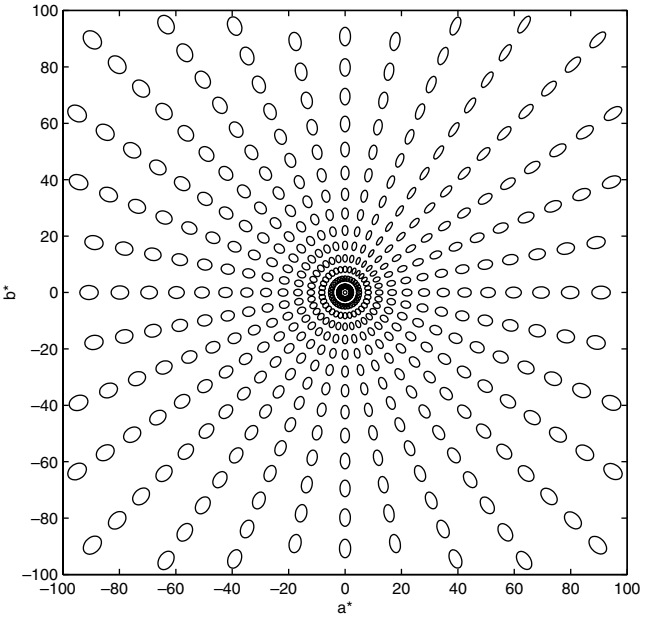


Figure 1.16 "Ellipses" corresponding to a color difference of $\Delta E_{CMC(1:1)} = 1.0$ in the a^*–b^* plane.

functions are independent of L^*. The figure, however, does not illustrate the effect of the weighting function S_L^{CMC}. The predominant trend seen in Figure 1.16 is the elongation of the "ellipses" as one goes radially outward from the center. A plot of similar "ellipses" for the ΔE^*_{ab} color difference would result in uniform circles of radius 1 throughout the plane. Thus, if the CMC formula is considered accurate, i.e., in better agreement with perceived differences, it indicates that the ΔE^*_{ab} color difference overemphasizes hue and chroma differences at higher chroma values. Visual color difference data verify that this is indeed the dominant deficiency in the uniformity of CIELAB.

While the CMC formula corrects for this dominant nonuniformity in CIELAB, because of its complexity and the number of terms, there are concerns about its statistical validity and generalization to data and situations beyond those for which it was originally derived.[19] In 1994, the CIE proposed a new color difference formula that incorporated only the more robust features of the CMC formula. The CIE-94 color difference formula[49] is obtained by using the following weighting functions in the difference equation of Equation 1.47:

$$S_C^{CIE94} = 1 + 0.045C^*_{ab} \tag{1.53}$$

$$S_H^{CIE94} = 1 + 0.015C^*_{ab} \tag{1.54}$$

The lightness weighting function $S_L^{CIE94} = 1.0$, and, in typical imaging applications, the parametric factors k_L, k_C, and k_H are all chosen to be unity. Asymmetric and symmetric versions of the formula have been defined. For the asymmetric formula, the chroma in Equations 1.53 and 1.54 corresponds to the chroma of the standard color, i.e., $C^*_{ab} = C^*_{ab,s}$. Just like the CMC formula, this implies that the distance between two samples is dependent on which one is chosen as the standard and which one as the sample. For the symmetric version of the formula, the chroma for the weighting functions in Equations 1.53 and 1.54 is defined as the geometric mean of the chromas of the sample and standard colors, i.e., $C^*_{ab} = \sqrt{(C^*_{ab,s}\, C^*_{ab,2})}$. This ensures that the distance between two samples is independent of which one is chosen as the standard. The color difference obtained using the above scaling functions in Equation 1.47 is referred to as the ΔE^*_{94} color difference.

The CIE-94 color difference formula is significantly simpler than the CMC formula. From the scaling functions of Equations 1.53 and 1.54, it is clear that the CIE-94 color difference formula scales down hue and chroma differences for higher chroma colors in comparison to CIELAB and thus corrects for the predominant deficiency in CIELAB. Along the L^* axis, the scaling factors are all unity, which ensures that (asymmetric) ΔE^*_{94} color differences about neutral colors that lie on the L^* axis are identical to the ΔE^*_{ab} Euclidean color difference. For chromatic colors, the weighting factors in Equations 1.53 and 1.54 are greater than unity, and the ΔE^*_{94} is therefore smaller than the ΔE^*_{ab} Euclidean color difference. A graphical visualization of the ΔE^*_{94} color difference formula is shown in Figure 1.17, where several elliptical closed curves are plotted in the a^*–b^* plane. These "ellipses" correspond to the loci of points whose color difference with respect to a standard color at the "center" of the "ellipse" is one ΔE^*_{94} unit (asymmetric formula). Once again, the central points have been excluded for clarity of the figure. The figure demonstrates trends similar to Figure 1.16, with "ellipses" elongated along the radial direction as one goes radially outward from the L^* axis and increasing in size along both dimensions. Due to the lack of a hue-dependent weighting term in the ΔE^*_{94} color difference formula, "ellipses" at the same radial distance from the origin in Figure 1.17 are congruent, unlike the corresponding "ellipses" in Figure 1.16. Through an integration of the chroma weighting function for the CIE-94 color difference formula, it is also possible to create an alternate color space in which the Euclidean distance is in very good agreement with the ΔE^*_{94} color difference formula.[244,295] Such a space is useful for visualizing ΔE^*_{94} color differences, as they correspond to the well-understood notion of Euclidean distance.

Recently, a new color difference formula that has been proposed for adoption as a CIE standard[179] is termed the CIE-DE2000 color difference formula, and the corresponding color difference is denoted as ΔE_{00}. The CIE-DE2000 color difference formula incorporates a hue-dependent weighting function S_H similar to the CMC formula. It also includes an additional term in the color difference that depends on the hue and chroma difference product that is motivated by the BFD formula (the R_T term in Equation 1.43).

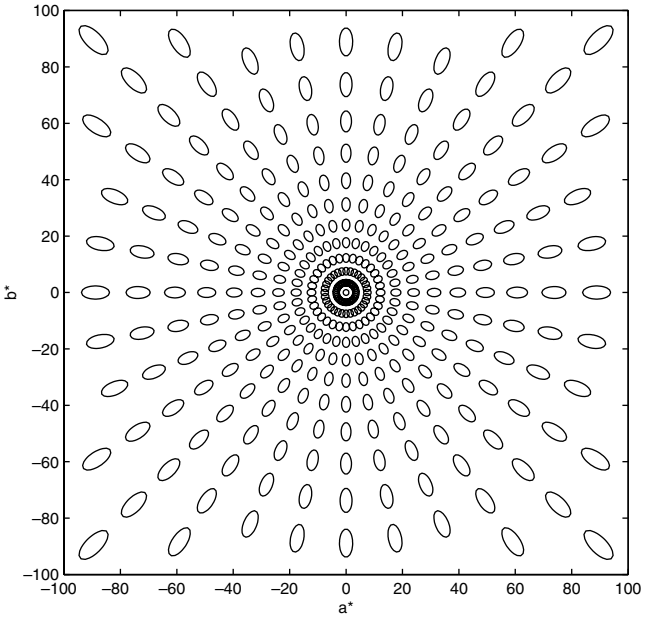


Figure 1.17 "Ellipses" corresponding to a color difference of $\Delta E^*_{94} = 1.0$ in the a^*–b^* plane.

The formula also includes a global rescaling of the a^* axis prior to computation of the hue and chroma. The CIEDE2000 color difference formula is designed to be symmetric by using averages of the standard and sample color values in the weighting functions. The resulting formula is fairly complex, and the reader is referred to Reference 179 for details. Figure 1.18 visually illustrates the behavior of the ΔE_{00} color difference formula in the a^*–b^* plane. The closed curves plotted in this figure correspond to the loci of points whose color difference, with respect to a standard color located roughly in the center of the curve, is one ΔE_{00} unit. The closed curves are well behaved in most regions and shaped like ellipses. In the blue region (around the negative b^* axis), however, the closed curves become non-convex and take on a very distorted shape. This is potentially problematic and probably does not agree with any psychophysical color difference data, most of which predict loci of visually equidistant color samples from a standard as convex almost elliptical closed curves. It would therefore be prudent to wait for a resolution of these issues before using the CIE-DE2000 color difference formula. In addition to the issue illustrated in Figure 1.18, additional concerns remain with regard to the CIE-DE2000.[170,180] Because the scaling functions have been derived from color difference datasets, there are questions about differences in the conditions under which the data were gathered and about the influence of the conditions on the scaling functions. In particular, some of the data might be influenced by lightness and chroma

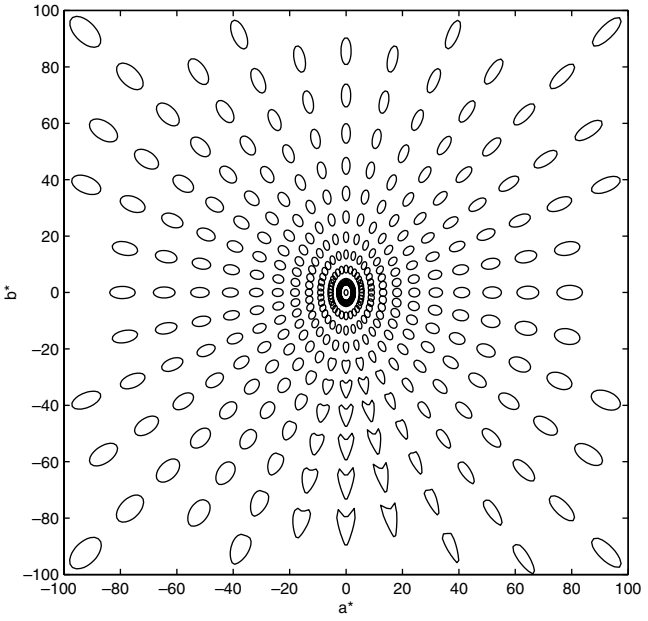


Figure 1.18 "Ellipses" corresponding to a CIEDE2000 color difference of $\Delta E_{00} = 1.0$ in the a^*–b^* plane.

crispening,† limiting the scenarios under which the formula is applicable.[170] In imaging applications, colors are typically surrounded by other similar colors, whereas visual datasets used for deriving the color difference formulae are based on comparisons of colors on a fixed background — normally mid-gray. For imaging applications, therefore, it is particularly important that the local lightness/chroma crispening in the experimental visual data should not influence the color difference formulae. In addition to the above concerns with color difference formulae, there are several fundamental questions associated with the CIELAB space itself.[169] Research on improved uniform color spaces and color difference formulae is therefore likely to continue.

1.8 Limitations of CIE colorimetry

CIE colorimetry as discussed above has several additional limitations beyond the deficiency in uniformity of the CIELAB and CIELUV color spaces that was already addressed above. First, variations among observers' color matching characteristics are not comprehended by the CIE standard observer and can result in a mismatch in colors for an actual color-normal observer, even when a metameric match is predicted by the standard observer. This is termed *observer metamerism*. While the extent of variation cannot be inherently reduced, it is useful to quantify it, because it forms a basis for deciding

† A description of lightness crispening can be found in the next chapter.

what color tolerances are allowable and the extent to which illumination has to be controlled to avoid illuminant metamerism (discussed in Section 1.5.2). Toward this end, the CIE has developed an additional concept of a *standard deviate observer,*[48] which allows users to establish confidence limits representing observer variability. The standard deviate observer was derived based on analysis of the variation in the original color matching data used to derive the CIE 1964 10° observer.

There is also significant debate about the accuracy of the standard CIE 1931 CMFs, given that these were derived from experimental data gathered almost a century ago, using equipment limited by the technology available at that time. In particular, the data used to generate the 2° standard observer[110,328] cannot be entirely reconciled with the data gathered later and used to generate the 10° standard observer.[284] Research has indicated some small but systematic deviations from the original CMFs in the short-wavelength regions.[78,278,285] It has been conjectured that these deviations may partly explain some of the blue hue nonlinearity in CIELAB that was discussed earlier. The CIE is also involved in an effort to derive new cone fundamentals (CMFs) for the 2° standard observer based on the more recent 10° observer data.[256,307] There are also fundamental questions about the connection between photometry and colorimetry and whether it is indeed possible to define the photometric (luminance) response in terms of the colorimetric response as one of the CMFs. Colorimetry therefore continues to be an area of active research. In addition to issues with established standards for color-normal observers, significant work is also required to better understand the "color" vision characteristics of color-anomalous observers.

Despite the above limitations, it is worth noting that the CIE system for colorimetry is a significant advance over methods for color specification prevailing before its adoption. The CIE system of colorimetry provides a methodology for the quantitative specification of colors (or at least, color correspondences). In this regard, the perception of color is significantly advantaged over other human senses such as touch and smell for which no system for quantitative specification currently exists and none is under imminent development.

1.9 Psychophysics of color

The human visual system as a whole displays considerable adaptation. It is estimated that the total intensity range over which colors can be sensed is around 10^8:1. While the cones themselves respond only over a 1000:1 intensity range, the vast total operating range is achieved by adjustment of their sensitivity to light as a function of the incident photon flux.[320] This adjustment is believed to be largely achieved through a feedback from the neuronal layers that provide temporal lowpass filtering and adjust the cones' output as a function of average illumination. A small fraction of the adaptation corresponding to a factor of around 8:1 is the result of a 4:1 change in the diameter of the pupil that acts as the aperture of the eye.[135(p. 23)]

Another fascinating aspect of human vision is the invariance of object colors under lights with widely varying intensity levels and spectral distributions. Thus, objects are often recognized as having approximately the same color in phases of daylight having considerable difference in their spectral power distribution and also under artificial illumination. This phenomenon is called *color constancy*. The term *chromatic adaptation* is used to describe the changes in the visual system that relate to this and other psychophysical phenomena.

While colorimetry provides a representation of colors in terms of three independent variables, it was realized early on that humans perceive color as having four distinct *hues* corresponding to the perceptually unique sensations of red, green, yellow, and blue. Thus, while yellow can be produced by the additive combination of red and green, it is clearly perceived as being qualitatively different from each of the two components. Hering[120] had considerable success in explaining color perception in terms of an opponent-colors theory, which assumed the existence of neural signals of opposite kinds, with the red–green hues forming one opponent pair and the yellow–blue hues constituting the other. Such a theory also satisfactorily explains both the existence of some intermediate hues (such as red-yellow, yellow–green, green–blue, and blue–red) and the absence of other intermediate hues (such as reddish-greens and yellowish-blues).

Initially, the trichromatic theory and the opponent-colors theory were considered competitors for explaining color vision. However, neither one by itself was capable of giving satisfactory explanations of several important color vision phenomena. In more recent years, these competing theories have been combined in the form of *zone theories of color vision*, which assume that there are two separate but sequential zones in which these theories apply. Thus, in these theories, it is postulated that the retinal color-sensing mechanism is trichromatic, but an opponent-color encoding is employed in the neural pathways carrying the retinal responses to the brain. These theories of color vision have formed the basis of a number of color appearance models that attempt to explain psychophysical phenomena. Typically, in the interests of simplicity, these models follow the theories only approximately and involve empirically determined parameters. The simplicity, however, allows their practical use in color reproduction applications involving different media where a perceptual match is more desirable and relevant than a colorimetric match.

A somewhat different but widely publicized color vision theory was the *retinex* (from *retin*a and cort*ex*) theory of Edwin Land.[174,173] Through a series of experiments, Land demonstrated that integrated broadband *reflectances* in red, green, and blue channels show a much stronger correlation with perceived color than the actual spectral composition of radiant light incident at the eye, or corresponding integrated radiances. He further postulated that the human visual system is able to infer the broadband reflectances from a scene through a successive comparison of spatially neighboring areas, which offers an alternate spatial form of adaptation similar to chromatic adaptation.

The *retinex* computation is one of the few models of vision that attempts to comprehend spatial interactions in color vision. Recent years have seen a resurgence of interest in the retinex theory.[203] Computational versions of the theory have recently been used, with some success, in the enhancement of color images,[145,146,239] in illuminant estimation for digital camera images,[97] gamut mapping of high dynamic range images,[281] and other applications.

One may note here that some of the uniform color spaces include some aspects of color constancy and color appearance in their definitions. In particular, both the CIELAB and CIELUV spaces employ an opponent color encoding and use white-point normalizations that partly explain color constancy. However, the notion of a color appearance model is distinct from that of a uniform color space. Typical uniform color spaces are useful only for comparing stimuli under similar conditions of adaptation and can yield incorrect results if used for comparing stimuli under different adaptation conditions. The CIE does not recommend the use of CIELAB and CIELUV spaces in conditions where the illuminant is "too different from average daylight."[47] Even under these restricted conditions, it is apparent from the discussion of Section 1.7.3 that, as appearance spaces, CIELAB and CIELUV are rather crude approximations. Color appearance models and spaces are the topic of the next chapter. A very brief outline is included here to provide a broad overview and establish the connections with the basic colorimetry and color difference formulae we have described here.

1.9.1 Chromatic adaptation and color constancy

Several mechanisms of chromatic adaptation have been proposed to explain the phenomenon of color constancy. Perhaps the most widely used of these in imaging applications is one proposed by von Kries.[167] He hypothesized that the chromatic adaptation is achieved through individual adaptive gain control on each of the three cone responses. Thus, instead of Equation 1.3, a more complete model represents the cone responses as

$$\mathbf{c}' = \mathbf{D}\mathbf{S}^T\mathbf{f} \tag{1.55}$$

where $\mathbf{D}$ is a diagonal matrix corresponding to the gains of the three channels, and the other terms are as before. The gains of the three channels depend on the state of adaptation of the eye, which is determined by preexposed stimuli and the surround, but independent of the test stimulus $\mathbf{f}$. This is known as the *von Kries coefficient rule.*

The term *asymmetric matching* is used to describe matching of color stimuli under different adaptation conditions. Using the von Kries coefficient rule, two radiant spectra, $\mathbf{f}_1$ and $\mathbf{f}_2$, viewed under adaptation conditions specified by the diagonal matrices $\mathbf{D}_1$ and $\mathbf{D}_2$, respectively, will match if

$$\mathbf{D}_1\mathbf{S}^T\mathbf{f}_1 = \mathbf{D}_2\mathbf{S}^T\mathbf{f}_2 \tag{1.56}$$

Thus, under the von Kries coefficient rule, chromatic adaptation can be modeled as a diagonal transformation for tristimuli specified in terms of the eye's cone responses. Usually, tristimulus values are specified not relative to the cone responses themselves but to CMFs that are linear transformations of the cone responses. In this case, it can readily be seen[335(p. 432)] that the tristimuli of color stimuli that are in an asymmetric color match are related by a similarity transformation[101] of the diagonal matrix $\mathbf{D}_1^{-1}\mathbf{D}_2$.

A von Kries transformation is commonly used in color rendering applications because of its simplicity and is a part of several standards for device-independent color imaging.[61,290] Typically, the diagonal matrix $\mathbf{D}_1^{-1}\mathbf{D}_2$ is determined by assuming that the cone responses on either side of Equation 1.56 are identical for white stimuli (usually a perfect reflector illuminated by the illuminant under consideration). The white-point normalization in CIELAB space was primarily motivated by such a model. Because the CIE XYZ CMFs are not *per se* the cone responses of the eye, the diagonal transformation representing the normalization is not a von Kries transformation and was chosen more for convenience than accuracy.[79]

In actual practice, the von Kries transformation can explain results obtained from psychophysical experiments only approximately.[335(pp. 433–451)] At the same time, the constancy of metameric matches under different adaptation conditions provides strong evidence for the fact that the cone response curves vary only in scale while preserving the same shape.[131(p. 15)] It therefore seems most likely that part of the adaptation lies in the nonlinear processing of the cone responses in the neural pathways leading to the brain.

A number of alternatives to the von Kries adaptation rule have been proposed to obtain better agreement with experimental observations. Most of these are nonlinear and use additional information that is often unavailable in imaging applications. Several of these are discussed in the next chapter, and additional information can be found in References 81 and 135 (pp. 81 and 217).

The phenomenon of color constancy suggests that the human visual system transforms recorded stimuli into representations of the scene reflectance that are (largely) independent of the viewing illuminant. Several researchers have investigated algorithms for estimating illuminant-independent descriptors of reflectance spectra from recorded tristimuli, which have come to be known as "computational color constancy algorithms."[88–90,93,195] Several of these algorithms rely on low-dimensional linear models of object and illuminant spectra, which will be discussed briefly in Section 1.11.5.1. A discussion of how these algorithms relate to the von Kries transformation rule and to human color vision can also be found in References 89, 90, and 322.

1.9.2 Opponent processes theory and color appearance models

The modeling of chromatic adaptation is just one part of the overall goal of color appearance modeling. While color appearance models are empirically determined, they are usually based on physiological models of color vision.

Most modern color vision models are based on "wiring diagrams" of the type shown in Figure 1.19. The front end of the model consists of L, M, and S (long, medium, and short wavelength sensitive) cones. The cone responses undergo nonlinear transformations and are combined into two opponent color chromatic channels (R-G and Y-B) and one achromatic channel (A). A positive signal in the R-G channel is an indication of redness, whereas a negative signal indicates greenness. Similarly, yellowness and blueness are opposed in the Y-B channel. The outputs of these channels combine to determine the perceptual attributes of hue, saturation, and brightness.

It is obvious that the above color-vision model is an oversimplification. Actual color appearance models are considerably more intricate and involve a much larger number of parameters, with mechanisms to account for spatial effects of surround and the adaptation of the cone responses, which was briefly discussed in the last section. Due to the immense practical importance of color appearance modeling to color reproduction systems, there has been considerable research in this area, and a standard color appearance model has been developed by the CIE.[50,181] Additional details of the state of the research in color appearance models can be found in the next chapter.

A common use of color appearance models in imaging applications is in reproducing images that are to be viewed under different viewing condi-

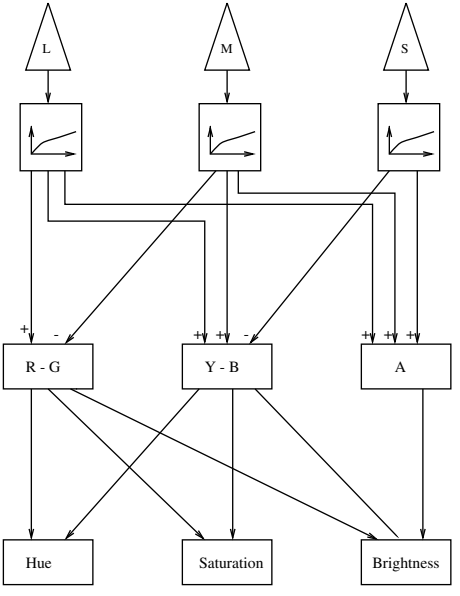


Figure 1.19 Typical "wiring diagram" for human color vision models (adapted from Reference 246).

tions from the original. An example of this situation would be the reproduction of a bright daylight-lit outdoor scene in a photograph/printed image that is to be viewed indoors under significantly lower light levels and illumination with relative SPDs very different from daylight. Matching the colorimetry of the outdoor scene under the indoor illumination is neither feasible nor desirable due to the very different states of adaptation of the eye under the differing conditions.[136] Instead, it is desirable to match the appearance as closely as possible. This is facilitated by color appearance models.

Figure 1.20 illustrates the conceptual use of the color appearance model in situations similar to those outlined above. For given colorimetry under (specified) reference viewing conditions, a color appearance model predicts the colorimetry required under the (specified) test viewing conditions for producing the same color appearance. To take a concrete example, if you view a reflective print under one illuminant and want to create a print that, when viewed under a different illuminant, produces an identical appearance (in your mind), the color appearance model gives you the colorimetry that your desired print should have. Note that the appearance that you are trying to match is that of the "original" print under the original viewing conditions and not the appearance of the original print under the new viewing illuminant. This is a common source of confusion in the use of color appearance models and chromatic adaptation transforms: it is incorrectly assumed that these models predict the color appearance of a physical color sample as it

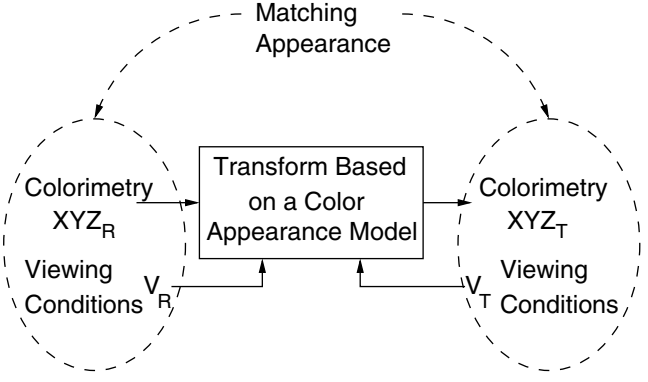


Figure 1.20 Use of a color appearance model to predict colorimetry under test viewing conditions that will match appearance of reference colorimetry under corresponding viewing conditions.

is moved across viewing illuminants. A color appearance model is not intended to predict the color appearance of a physical object under a different "test" viewing illuminant from its colorimetry under a "reference" viewing illuminant. Clearly, if we have metameric samples that match under the "reference illuminant," a color appearance model will indicate that they match under all test illuminants whereas, in practice, we know this is not true. Color appearance models only model the adaptation process in the eye — they cannot and do not model the physical changes in the viewing illuminant; these can be properly accounted for only by using spectral measurements using spectrophotometers and spectroradiometers.

Despite the statements in the above paragraphs, arguments from color constancy can be used to say that, for most objects, color appearance models would provide a good approximation to the colorimetry of the original object under the new illuminant. However, because color constancy holds only approximately, so does this argument, and it is important to understand that the failure of a color appearance model in predicting the change in appearance of a physical object when it is moved from one viewing illuminant to another is an instance of incorrect use of the model rather than a failure of the model.

1.10 Spatial characteristics of color vision

The CIELAB color space and associated color difference formulae try to account for the nonlinearities in the color sensing process for the comparison of relatively large uniform color regions and do not account for any spatial interactions in the eye. Because the eye acts as a spatial lowpass filter that can average over high spatial-frequency differences in images, a point-by-point computation of color differences in CIELAB color space is not appropriate for the comparison of images. Though the eye exhibits lowpass characteristics for both luminance† and chrominance spatial information, the bandwidth for the chrominance channels is much lower than that of the luminance channels. This fact has been exploited in numerous applications, including color TV, where the transmission bandwidth allocated to the chrominance channels is significantly lower than that for the luminance channel.

Several researchers have modeled the spatial properties of the eye as a linear filter and studied the frequency response of the eye for luminance and chrominance spatial patterns.[87,95,103,164,345] However, a complete model for human vision that is perceptually uniform and incorporates the effects of the spatial interactions is yet to be developed. While sophisticated models that explain a large number of psychophysical effects have been developed for luminance information,[66,178] their extensions and the development of alternative models that are capable of handling color data are still an active

† Strictly speaking, the luminance response of the eye is believed to be the bandpass but, for all practical purposes, the low-frequency attenuation can be ignored.

area of research.[331] Several of the models in development are quite complicated and incorporate features designed to explain a wide variety of psychophysical effects.[73,346]

While sophisticated models are necessary in some cases,[67] in several imaging applications significant improvements can be made by incorporating relatively simple spatial models of human color vision. Computational simplicity is also a necessity in situations where the models are used in iterative loops for the optimization of image processing algorithms. As a consequence, several simplified models have been developed that model the spatial characteristics of color vision as simple luminance/lightness and chrominance contrast sensitivity functions. One of the popular models in this class of simplified models is the spatial extension of the CIELAB (S-CIELAB) model,[345] which will be used here for illustration of some of the features of these models.

The S-CIELAB model first transforms a color image into three opponent color image planes, O_1, O_2, and O_3, corresponding to black–white (*luminance*), red–green, yellow–blue image components, respectively. These three color coordinates are defines as a linear transformation of the CIE XYZ tristimulus values as

$$\begin{aligned} O_1 &= 0.279X + 0.72Y - 0.107Z \\ O_2 &= -0.449X + 0.29Y - 0.077Z \\ O_3 &= 0.086X + 0.59Y - 0.501Z \end{aligned} \tag{1.57}$$

Each opponent-colors image is convolved with a kernel filtered by two-dimensional separable spatial kernels consisting of mixed Gaussians

$$f^j(x,y) = k^j \sum_i w_j^j E_i^j(x,y) \tag{1.58}$$

$$E_i^j(x,y) = k_i^j \exp\left(\frac{-(x^2+y^2)}{(\sigma_i^j)^2}\right) \tag{1.59}$$

where $f^1(x,y)$, $f^2(x,y)$, and $f^3(x,y)$ are the kernels for O_1, O_2, and O_3, respectively. Variables x and y denote the spatial dimensions, and the standard deviation σ_i^j determines the spatial spread of the corresponding Gaussian in the mixture. In the discrete implementation, the scale factor k_i^j is chosen so that $E_i^j(x,y)$ sums to 1 over the spatial extent along x and y. The scale factors k^j are chosen so that, for each color plane, its two-dimensional kernel $f^j(x,y)$ sums to one. The parameters (w_i^j, σ_i^j) for the three opponent-colors are listed in Table 1.2, where σ_i^j are in degrees of visual angle.

Table 1.2 Parameters for the s-CIELAB HVS Model

	w_i^j	σ_i^j
Luminance O_1 (i = 3)	0.921	0.0283
	0.105	0.133
	–0.108	4.336
Red-green O_2 (i = 2)	0.531	0.0392
	0.330	0.494
Blue-yellow O_3 (i = 2)	0.488	0.0536
	0.371	0.386

In the S-CIELAB model, the opponent color images O_1, O_2, and O_3 are spatially filtered through the corresponding spatial kernels $f^1(x,y)$, $f^2(x,y)$, and $f^3(x,y)$ to obtain images O_1', O_2', and O_3', respectively.

$$O_j'(x, y) = (O_j(x, y) * f^j(x, y)) \qquad j = 1, 2, 3 \tag{1.60}$$

where * denotes convolution. The filtered images O_1', O_2', and O_3' are then transformed back to CIE XYZ coordinates using the inverse of the transformation in Equation 1.57. The CIE XYZ coordinates are then transformed into CIELAB using the transformation described earlier in Section 1.7.2. The resulting image is the S-CIELAB representation of the original image. To evaluate the differences between two images,† the images are converted into their S-CIELAB representations, and pixel-wise color differences are computed between them in the CIELAB space, producing a color difference image. Typically, the CIE ΔE^*_{ab} color difference formula has been used, although the other difference formulae based on CIELAB described in Section 1.7.4 could also be used. The color difference image obtained from this process represents a spatial map of the visual difference or distortion between the images.[342] The distortion map may be used directly to determine regions where differences will be perceived, or the error over the image pixels may be averaged to determine a single measure of the perceived difference between the images.

The spatial filter kernels for S-CIELAB corresponding to the parameter values of Table 1.2 are shown in Figure 1.21a, b, and c for the black–white, red–green, and yellow–blue opponent channels, respectively. The shapes of the kernels clearly indicate the greater detail resolution for the black–white (*luminance*) channel in comparison to the red–green and yellow–blue chrominance channels, which is designed to mimic the behavior of the human visual system. Because the individual kernels sum to unity, filtering a large uniform region does not change its values, and the color difference computation

† It is assumed that the two images are registered well with each other. The S-CIELAB metric is not meaningful for misregistered images.

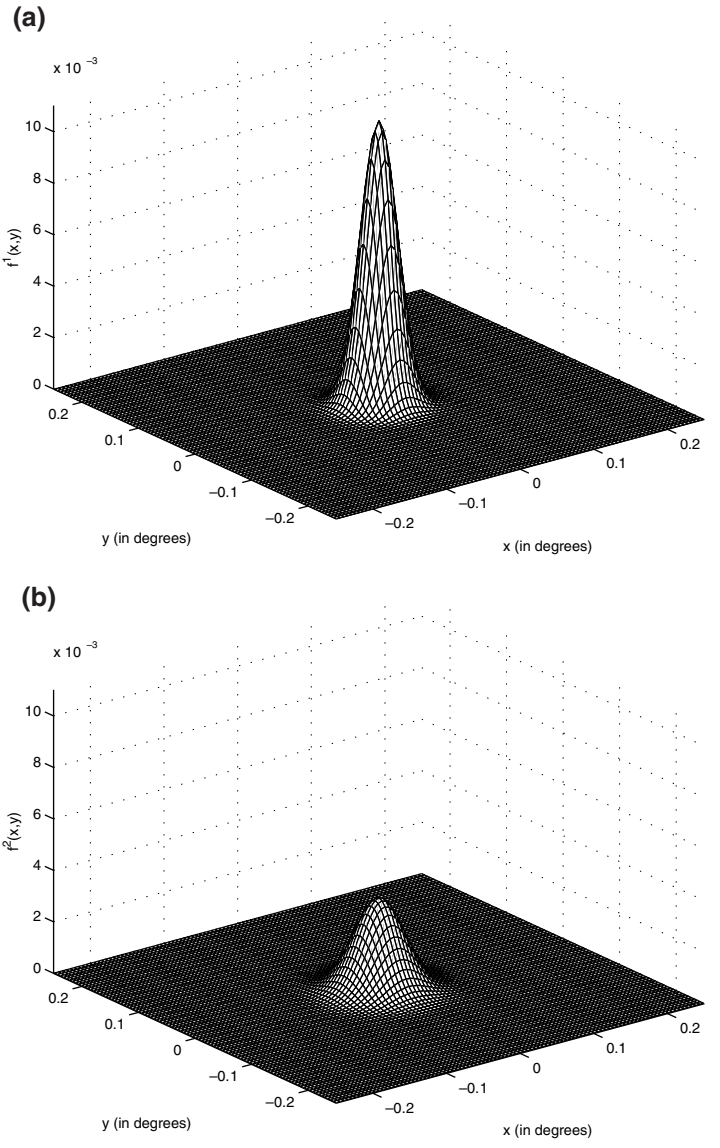


Figure 1.21 Spatial convolution kernels for the S-CEILAB opponent color channels: (a) $f^1(x,y)$, black–white; and (b) $f^2(x,y)$, red–green (continues).

reduces to the direct computation of color difference between the original images in CIELAB. Thus, the S-CIELAB model agrees with CIELAB for large uniform regions, which is desirable, as CIELAB is already a widely used model for computing color differences between spatially uniform regions of differing color.

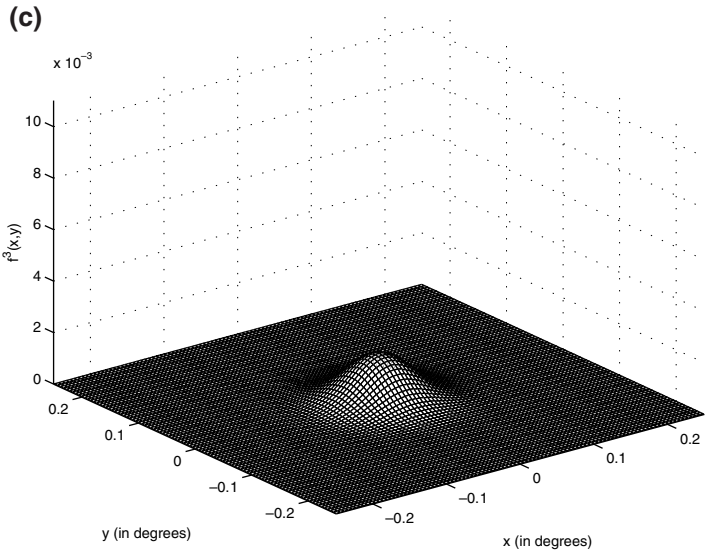


Figure 1.21 (continued) Spatial convolution kernels for the S-CEILAB opponent color channels: (c) $f^3(x,y)$, yellow–blue.

The S-CIELAB model is attractive because of its simplicity and close association to the well-accepted CIELAB standard. It has therefore been used recently in several different applications.[10,342–344] Several other models with similar characteristics have also been employed successfully in color imaging applications.[87,95,103,164] These models provide varying degrees of agreement with psychophysical data and few, if any, comprehensive objective tests of the different models exist. Research on improving the models and on the development of new models is also continuing.

1.11 Color image reproduction and recording devices

In the physical world, color images exist as spatially varying spectral radiance or reflectance distributions. To process these images digitally, the images must be sampled both spatially and spectrally. The issues involved in spatial sampling and reconstruction of images have been discussed at length in signal processing literature and will not be repeated here. The aspects of spectral sampling and color recording for images are also important, and these are addressed here. The reproduction of color images works in the opposite direction from the recording process. The physical realization of color images from recorded information requires synthesis of spatially varying spectral radiance or reflectance distributions. This section provides an overview of common color reproduction devices, the spectral characteristics of natural images and images from these reproduction devices, and methods for recording these images. Color reproduction devices are discussed first,

because color recording systems may also be used to record color reproductions and may exploit the characteristics of the reproduction device.

1.11.1 Color output systems

Nature provides a variety of mechanisms by which color may be produced. As many as 15 distinct physical mechanisms have been identified that are responsible for color in nature.[210] While only a fraction of these mechanisms are suitable for technological exploitation, there is still considerable diversity in available technologies and devices for displaying and printing color images.

Color output devices can be classified broadly into two main types: *additive* and *subtractive*. Additive color systems produce color on a dark background through the combination of differently colored lights, known as primaries. The qualifier additive is used to signify the fact that the final spectrum is the sum (or average) of the spectra of the individual lights, as was assumed in the discussion of color matching in Section 1.4. Typically, the additive primaries are red, green, and blue (RGB). The additive mixing of RGB primaries is illustrated in Figure 1.22. The combination of red and green forms yellow, of red and blue forms magenta, and of blue and green forms cyan. Combination of all three primaries at full intensities produces white. Intermediate colors are obtained by varying the individual primary intensities. Examples of additive color systems include color cathode ray tube (CRT) displays and projection video systems.

Color in subtractive systems is produced through a process of removing (subtracting) unwanted spectral components from "white" light. Typically,

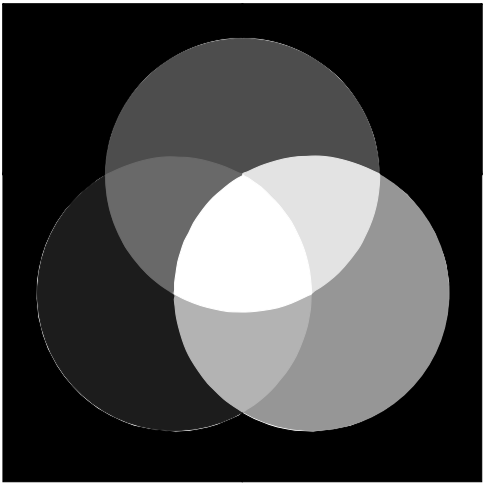


Figure 1.22 ***(See color insert following page 430)*** Additive mixing of red, green, and blue primaries.

such systems produce color on transparent or reflective media, which are illuminated by "white" light having spectral components distributed over the visible spectrum. By overlaying these media with colorants that selectively absorb light of certain wavelengths while transmitting other wavelengths, different colors are produced. Typical subtractive systems are based on cyan, magenta, and yellow (CMY) colorants that absorb light in the red, green, and blue spectral regions, respectively. The red, green, and blue spectral regions are roughly defined as the intervals 600–700, 500–600, and 400–500 nm, respectively. Each colorant absorbs its complementary color and transmits the rest of the visible range of the spectrum. The process is illustrated in Figure 1.23. The individual CMY colorants eliminate RGB spectral regions, respectively. The combination (overlay) of cyan and magenta eliminates both red and green, producing blue; the combination of cyan and yellow eliminates red and blue, producing green; and the combination of magenta and yellow eliminates green and blue, producing red. The combination of the maximum amounts of all three produces black. Intermediate colors are produced by varying the colorant amounts. Dye sublimation printers, color photographic prints, color slides, and halftone color printers are representatives of the subtractive process. Halftone color printing, which is commonly used for lithographic/electro-photographic printing and in most desktop inkjet color printers, may also be viewed as a *hybrid* system,[266] since the colorants combine subtractively, but the perceived color is the average of the differently colored regions over a small area.

Any practical output system is capable of producing only a limited range of colors. The range of producible colors on a device is referred to as its *gamut*. Red, green, and blue primaries are chosen for additive systems,

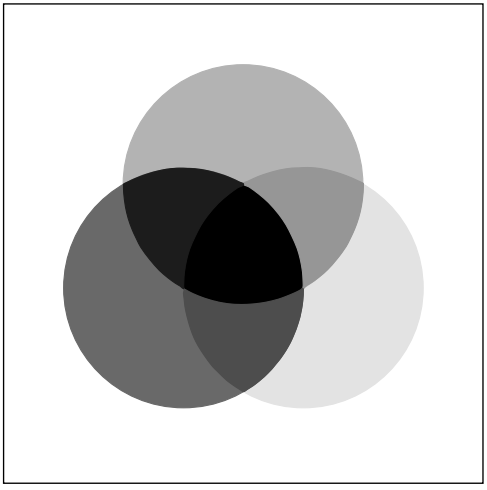


Figure 1.23 ***(See color insert)*** Subtractive combinations of cyan, magenta, and yellow colorants.

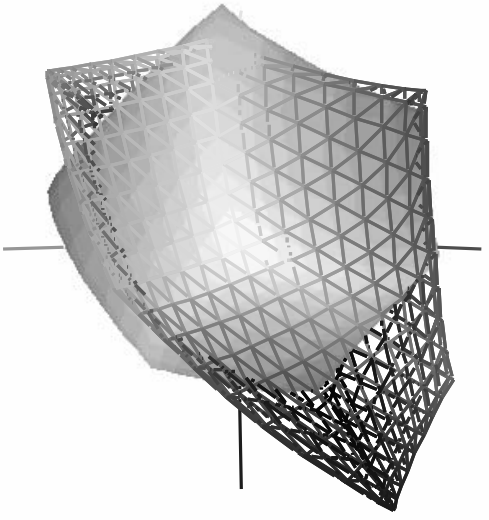
because they offer the largest possible gamut; for the same reason, subtractive systems use cyan (C), magenta (M), and yellow (Y) colorants. In subtractive systems, it is also common to use a fourth black (K) colorant that absorbs light (almost) uniformly over the visible spectrum. This improves the gamut by allowing darker colors to be produced and also allows better reproduction of achromatic (gray) colors and lower cost through the replacement of the more expensive CMY colorants with the K. An excellent description of additive and subtractive color reproduction and additional reasons behind the choice of primaries can be found in Reference 341, Chapter 3.

The gamut of a device can be represented by a three-dimensional solid in any color space such as CIEXYZ/CIELAB. Because two-dimensional representations are more convenient for display, it is common to utilize a CIE xy chromaticity diagram for this purpose. On the CIE xy chromaticity diagram, the gamut of an additive system appears as a convex polygon with the primaries representing the vertices. For the usual case of three red, green, and blue primaries, the gamut is a triangle on the CIE xy chromaticity diagram. Because most subtractive and hybrid systems are nonlinear, their gamuts have irregular shape and are not characterized by such elegant geometric constructs. The two-dimensional representation of the gamut on the CIE xy diagram presents only an incomplete (and difficult-to-interpret) picture, because it does not represent the full three-dimensional data. With the increase in computing speeds and advances in computer graphics, visualization techniques are now used to render three-dimensional views of the gamuts.[154,251] The ability to manipulate these views interactively is extremely useful in understanding the capabilities and limitations of the different color output devices. Three different views comparing the gamut of a CRT monitor and the gamut of a dye-sublimation continuous tone color printer are shown in Figure 1.24, and identical views for an inkjet halftone color printer are shown in Figure 1.25. In both cases, the wire frames represent the gamut of a CRT monitor, and the solids represent the gamuts of the printer.† These views demonstrate that the gamuts of these three output devices are fairly different, with several colors that can be produced on one device and not on another. Overall, the gamut of the monitor is the largest, followed by the gamut of the continuous tone printer and then by the inkjet halftone printer, which has a rather small gamut in comparison to the other devices. This mismatch in the gamut between the devices poses significant challenges in cross-media color reproduction and is discussed in some detail in Chapter 10, on gamut mapping.

To discuss colorimetric reproduction on color output devices, it is useful to introduce some terminology. The term *control values* is used to denote signals that drive a device. The operation of the device can be represented as a multidimensional mapping from control values to colors specified in a device-independent color space. This mapping is referred to as the (device)

† The gamuts displayed here are computed assuming that no flare is present when viewing the CRT display. Flare, i.e., ambient external room light reflected by the CRT display screen, can cause a significant reduction in the effective gamut of a CRT display.[263]

(a)



(b)

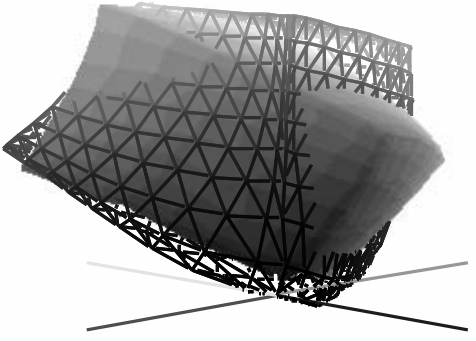


Figure 1.24 ***(See color insert)*** Comparison of a CRT monitor gamut (shown as a wire frame) and a continuous-tone dye-sublimation printer gamut (shown as a solid) in CIELAB space: (a) top view along the L^* axis, and (b) a perspective projection (continues).

(c)

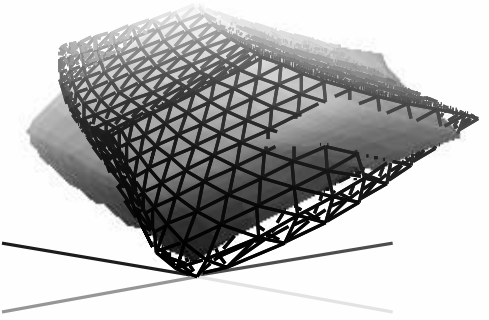


Figure 1.24 (continued) ***(See color insert)*** Comparison of a CRT monitor gamut (shown as a wire frame) and a continuous-tone dye-sublimation printer gamut (shown as a solid) in CIELAB space: (c) another perspective projection.

forward device transform. interactively specified colors in a device-independent color space must be mapped to device control values to obtain colorimetric output, it is necessary to determine the inverse of the multidimensional forward device transform. The determination of the forward device transform and the inverse transform necessary for mapping device independent colors to device control values is sometimes called device *profiling*.

If the device's operation can be accurately represented by a parametric model, the profiling is readily done by determining the model parameters from a few measurements. If no useful model exists, a purely empirical approach is necessary in which the forward device transform is directly measured over a grid of device control values. The inversion may be performed in a closed form if the characterization uses a device model that allows this. If an empirical approach is employed in characterization, or if the model used is noninvertible (often the case with nonlinear models), one has to resort to numerical methods in the inversion step.

1.11.1.1 Cathode ray tubes

The most widely used display device for television and computer monitors is the color CRT. The CRT produces visible light by bombardment of a thin

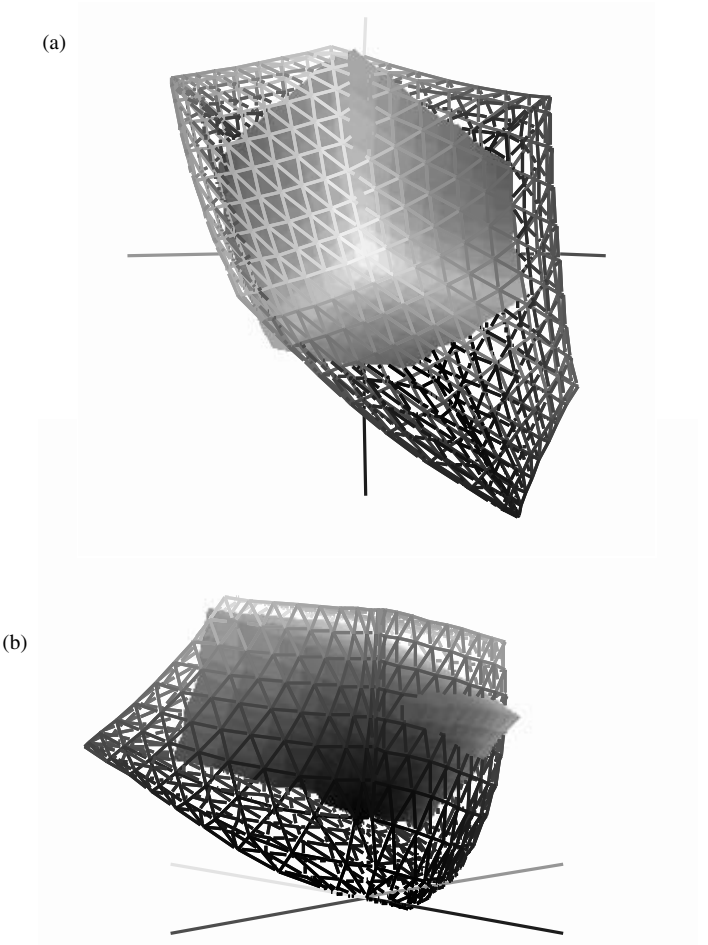


Figure 1.25 ***(See color insert)*** Comparison of a CRT monitor gamut (shown as a wire frame) and an inkjet printer gamut (shown as a solid) in CIELAB space: (a) top view along the L^* axis and (b) a perspective projection (continues).

layer of phosphor material by an energetic beam of electrons. The electron beam causes the phosphor to fluoresce and emit light whose spectral characteristics are governed by the chemical nature of the phosphor. The most commonly used color CRT tubes are the shadow-mask type, in which a mosaic of red, green, and blue light-emitting phosphors on a screen is illuminated by three independent electron beams. The intensity of light emitted by the phosphors is governed by the velocity and number of electrons. The beam is scanned across the screen by electrostatic or electromagnetic deflection mechanisms. The number of electrons is modulated in synchronism with the scan to obtain spatial variations in the intensity of the light emitted by the three phosphors. At normal viewing distances, the light

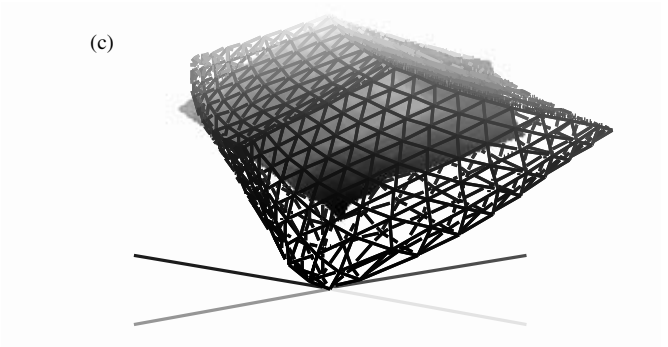


Figure 1.25 (continued) ***(See color insert)*** Comparison of a CRT monitor gamut (shown as a wire frame) and an inkjet printer gamut (shown as a solid) in CIELAB space: (c) another perspective projection.

from the mosaic is spatially averaged by the eye, and the CRT thus forms an additive color system.

There are several design choices in the manufacture of shadow mask CRTs. Other competing designs offer improved resolution by utilizing a layered phosphor instead of a mosaic. The reader is referred to References 37 and 323 for a description of the different technologies and involved tradeoffs. A detailed description of physical principles involved in the operation of these devices is provided in Reference 272 (pp. 79–200).

Color in CRT displays is controlled through the application of different voltages to the red, green, and blue guns. For a complete colorimetric characterization of these devices, the CIE XYZ tristimulus values (or other tristimuli) must be specified as a spatially varying function of the voltages applied to the three guns. A brute-force approach to this problem, using a multidimensional lookup table, is not feasible because of the extremely large number of measurements required. Hence, simplifying assumptions should be made so as to make the problem tractable.

Assumptions of spatial uniformity, gun independence, and phosphor constancy are commonly made to simplify CRT colorimetry.[29] Spatial uniformity implies that the color characterization of the CRT does not vary with position. Gun independence refers to the assumption that the three phosphors and their driving mechanisms do not interact. This implies that the incident intensity at the eye when the guns are operated simultaneously is the sum of the intensities when the guns are operated individually. Phosphor constancy refers to the assumption that the relative spectral power distribution of light emitted by the phosphors does not change with driving voltage (i.e., at all driving voltages, the spectra emitted by a phosphor are scalar multiples of a single spectrum).

With the above three assumptions, the problem of characterizing the CRT reduces to a problem of relating the intensities of the individual red, green, and blue channels to their corresponding gun voltages. The problem

can be further simplified through the use of a parametric model for the operation of the individual guns. Typically, these models are based on the exponential relation between the beam current and grid voltage in vacuum tubes.[43,175] For each gun, the spectrum of emitted light in response to a control voltage, v, is modeled by an expression of the form[23,222]

$$\left((1-\beta)\frac{v}{v_m}+\beta\right)^{\gamma} h(\lambda) \tag{1.61}$$

where v_m = maximum value of the voltage
$h(\lambda)$ = emitted phosphor spectrum at the maximum voltage
β = an offset
γ = the exponential parameter

For appropriate setup of the monitor offset and brightness controls,[23] the offset term $\beta = 0$ and the relation simplifies to $(v/v_m)^{\gamma} h(\lambda)$. The exponent, γ, is commonly referred to as the *monitor gamma* and is normally around 2.2 for most color monitors. Because the power-law parametric model is only approximate, several modifications of it have been used by researchers.[21,23,64,99] In practice, the addition of the offset term as illustrated in Equation 1.61 offers a significant improvement, because it accounts for deviations from the pure power-law behavior due to differences in the brightness control setting for the CRT.[21,23,263] Using the parametric models, CRT monitors can be readily characterized using a small number of measurements.

The CRT phosphors define a set of additive primaries. If the CMFs corresponding to these primaries are used in color specification, they can be directly used to drive the electron guns if the signals are precorrected for the power-law nonlinearity mentioned above. The transformation from CIE XYZ tristimulus values is given by a linear transformation, corresponding to a transformation from the CIE primaries to the phosphor primaries, followed by a one-dimensional transformation that is determined by the parametric model used to represent the operation of the individual electron guns.[23] Typically, this operation involves exponentiation to the power of $1/\gamma$ and is known as *gamma correction*. As mentioned in Section 1.5.4, TV signals are normally gamma corrected before transmission. One may note here that quantization of gamma corrected signals results in wider quantization intervals at higher intensities where the sensitivity of the eye is also lower. Therefore, just like speech companding, gamma correction of color tristimuli prior to quantization in a digital system (or transmission in a limited bandwidth system) reduces the perceptibility of errors and contours in comparison to a scheme in which no gamma correction is used.[130(p. 393),163,166,227,232]

For colors that the phosphors are capable of producing, fairly good color reproduction can be obtained on a CRT using the models mentioned above. Using only a few measurements of the individual channel responses, the models provide a very good approximation to the actual display. Typical color errors[23,263] from the offset-gamma model described above are an aver-

age ΔE^*_{ab} under 1 and a maximum ΔE^*_{ab} around 2, and the corresponding numbers in ΔE^*_{94} units are 0.5 and 1.6, respectively. In comparison to other color reproduction devices, these color errors are extremely small. CRTs have therefore been often used in color vision research where a very high color accuracy is required. In practical color imaging applications, the limitations in the CRT color gamut due to the phosphors used cause significant color errors for colors that lie beyond the gamut. This is one of the primary sources of color errors seen in broadcast TV.

The assumptions of gun independence and phosphor constancy have been tested by several researchers and found to hold reasonably well.[21,29,64,65,263] However, in most CRT monitors, for the same driving voltage, the light intensity is brightest at the center and falls off toward the edges. The change in luminance over the screen can be as high as 25%.[196(p.104)] Therefore, the assumption of spatial uniformity does not strictly hold. Because the eye's sensitivity itself is not uniform over the entire field of view, and because the eye adapts well to the smooth variation in intensity across the screen, the spatial nonuniformity of CRTs is not too noticeable.† An algorithm for correcting for spatial inhomogeneity is discussed in Reference 63.

1.11.1.2 LCD displays

Liquid crystal display (LCD) flat panels are becoming increasingly common as computer color displays due to their compact size and low power consumption. These displays are now available at increasingly higher spatial resolutions and in larger screen sizes with image quality that meets or exceeds that of typical cathode-ray-tube (CRT) displays.[327] While the market for CRTs continues to grow at present, in the long run, flat panel displays are expected to replace CRTs as the primary computer displays.[38,321]

With widespread use, there is also an increased need for color management for LCD displays, which enables accurate control of color in displayed images. While the color characteristics of CRT displays and methods for their color calibration have been extensively studied and reported, the color characteristics of LCD displays and methods for calibration have come to the forefront only in the last few years and have received only limited attention in published literature. A recent comparison of the color characteristics and gamut considerations for LCD displays vs. CRTs can be found in Reference 263.

The most common LCD displays for computers are back-lit active matrix LCDs (AMLCDs) of the "twisted nematic" type.[162(p. 72)] These are manufactured by deposition and patterning of (active) pixel electronics on a glass substrate. Each pixel element consists of a pair of linear polarizers with liquid crystal (LC) material sandwiched in between. Figure 1.26 graphically illustrates a pixel element. The two linear polarizers are orthogonally oriented;

† The imperceptibility of the slow lightness variation over a CRT screen can also be explained in terms of the bandpass nature of the eye's lightness/luminance spatial response, which severely attenuates the perceptual impact of this variation.

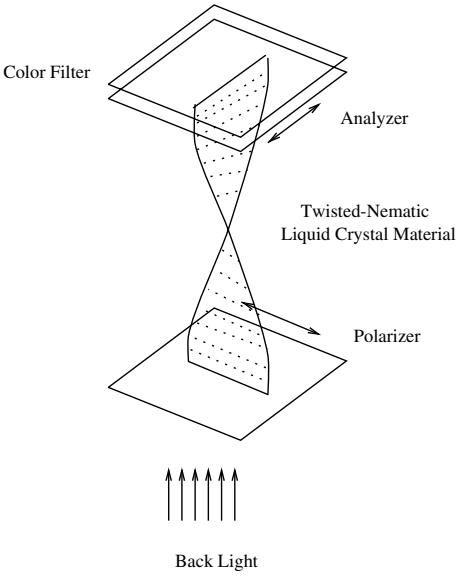


Figure 1.26 Structure of an LCD display pixel.

light does not pass through the display except for actions of the LCs. The surfaces adjacent to the LC molecules are typically designed so that (in the absence of any electric field) the LC molecules align in a 90° twisted configuration, which rotates the plane of polarization of incident linearly polarized light by a 90° angle.[291(pp. 429–430)] The "input" polarizer on the backside polarizes the light coming from the lamp behind the display. This polarized light encounters the LC molecules, which rotate its plane of polarization by 90°, allowing it to pass through the output polarizer, resulting in an ON pixel. The pixel is turned OFF by the application of an electric field. Due to their *dielectric anisotropy*, the LC molecules tend to align with the electric field and move away from their twisted state. In a strong enough electric field, the molecules are almost completely aligned with the electric field. This causes the pixel to be turned off, as the LC molecules no longer produce the 90° rotation in the plane of polarization that is required for the output polarizer to transmit the light.

Color displays are produced by laying a mosaic of red, green, and blue colored filters on the substrate glass aligned with the pixel array. Quite often, the individual RGB pixels are rectangular and arranged so that three horizontally adjacent rectangular RGB pixels constitute a single square "color pixel" (which is set farther away from other "color pixels" in comparison to the spacing between the individual RGB pixels). The display thus appears to be composed of stripes of rectangular RGB pixels going vertically across the screen. The backlight is typically a fluorescent lamp with three prominent peaks in the red, green, and blue regions of the spectrum.

In most AMLCD color displays, the RGB pixels are driven and controlled independently. The emitted light is combined and averaged in the eye (just as for CRTs). Therefore, the three RGB channels combine through simple addition of light, and one can expect channel independence to hold for these displays. This allows a complete characterization of the display from a per-channel characterization. Furthermore, if the switching mechanism of the LCD pixel cell described above is spectrally nonselective (i.e., when a pixel is driven by applying a voltage the percent change in spectral transmittance is the same across all wavelengths), the channel-chromaticity-constancy assumption of also applies, further simplifying the characterization to a determination of the individual channel intensity or tone response curves. The tone-response curves for the individual channels then correspond to what is commonly referred to as the *electro-optic response,* in LCD terminology. The electro-optic response of an LCD pixel cell (for on-axis viewing) tends to be an S-shaped curve (like sigmoidal functions).[162] This response is quite different from the power-law relation for CRTs and cannot readily be modeled by the parametric power-law relation of Equation 1.61. Additional measurements are therefore typically required for the characterization of LCD displays and alternate parametric relations have also been proposed.[172]

Because LCD displays were first deployed in an environment dominated by CRTs, the color filters for these devices were designed to have the same chromaticities as the CRT primaries. This allows the devices to be directly driven by the same signals as those used for CRTs, with only one-dimensional compensations for the per-channel tone-response curves. Often, displays incorporate built-in, switchable per-channel compensation curves that could change the actual observed tone–response curves for the display. These compensation curves are also typically set up to effectively mimic a power-law relation[84] between the driving signals and the observed luminance.

Using the assumption of channel independence, LCD displays can be color profiled with good accuracy. Average and maximum color errors in ΔE^*_{ab} units are around 2.4 and 4.5 with corresponding numbers in ΔE^*_{94} units around 1 and 2, respectively.[263] This level of error is well below the acceptable range for most imaging applications, although use in color science experimentation would require greater accuracy.

A common problem with most AMLCD displays is one of limited viewing angle. Due to the birefringent nature of liquid crystal (LC) molecules, polarized light entering an LC material off axis is treated differently from polarized light entering along the optical axis (0° angle of incidence wrt to the normal). The LC molecules are less effective in performing the proper rotation for polarized light that enters the LC material off axis. The electro-optic transfer function of LCDs therefore tends to be angle dependent.[291(p. 430)] In addition, the LCs are dispersive and operate differently on different wavelengths of light, especially when responding to off-axis incident light. Thus, as one moves off axis (either up or down or from side to side), significant variations in hue and contrast occur with the change in

viewing angle.[91] As one moves far enough off axis, the contrast is diminished to the point that it limits the effective viewing angle.

A change in viewing angle of 30° for an AMCLD display can result in average color shifts of over 5 ΔE^*_{ab} units.[263] Figure 1.27 shows color shifts observed in an experimental display for a change in viewing angle from 0 to 30° projected on the a^*–b^* plane. The lines in this plot represent the color shifts with the "*" representing the color seen along a 30° viewing angle and the other end representing the color seen along a 0° viewing angle. As the viewing angle changes from 0 to 30°, the colors move toward the center of the a^* and b^* plane. Thus, the color shifts are such that colors tend to desaturate as the viewing angle increases. The predominant effect seen in off-axis viewing is a reduction in contrast and saturation. The large magnitude of the differences indicates that such an LCD display should be used only for a limited viewing angle if accurate color is desired.

Several innovative techniques have been developed to increase this useful viewing-angle range.[119,15,161,230] Recent commercial displays include alternative technologies such as *in plane switching* to provide an improved usable angle of viewing.[197]

1.11.1.3 *Contone printers*

*Con*tinuous *tone* printers are subtractive color devices. They use different concentrations of the cyan, magenta, and yellow colorants to control the

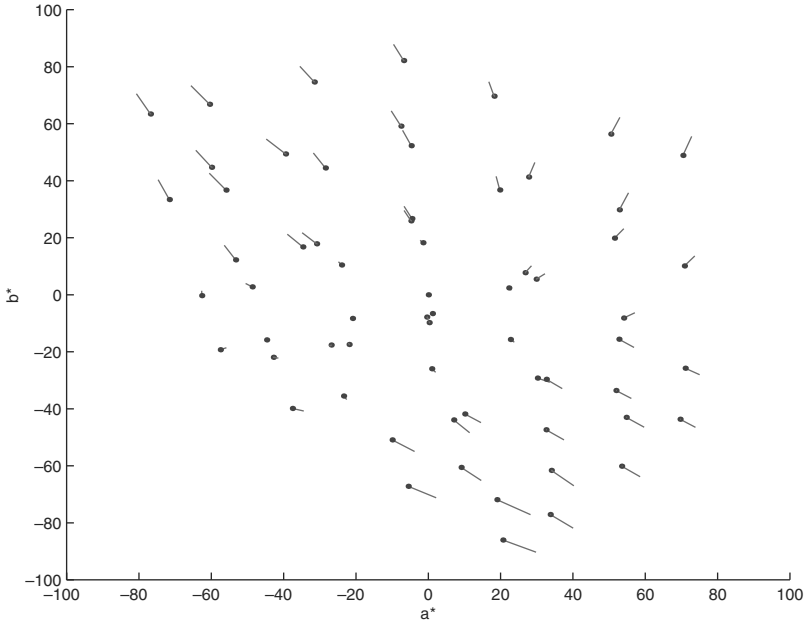


Figure 1.27 Color shifts in a^* and b^* for the LCD display test patches for a change in viewing angle from 0 to 30°.

absorption in the red, green, and blue regions and thereby produce different colors. The term *continuous* comes from the fact that, at each spatial location, these printers vary the colorant concentrations or amounts over a continuous range (in contrast with halftone printers, which are discussed next).

Most subtractive color reproduction systems are inherently nonlinear and cannot be modeled as easily or accurately as additive systems. The subtractive principle is schematically shown in Figure 1.28 for a transmissive system. If the incident light spectrum is $l(\lambda)$, the spectrum of the light transmitted through the three layers is given by $g(\lambda) = l(\lambda)t_1(\lambda)t_2(\lambda)t_3(\lambda)$, where $t_i(\lambda)$ is the spectral transmittance of the ith layer. If the colorants are transparent (i.e., do not scatter incident light), and their absorption coefficients are assumed to be proportional to their concentration (Bouguer–Beer law), it can be shown that [108(Chap. 7)] the *optical density* of the ith colorant layer, which is defined as the negation of the logarithm (base 10) of its transmittance, is given by

$$d_i(c_i; \lambda) = -\log_{10} t_i(c_i; \lambda) = c_i d_i(\lambda) \tag{1.62}$$

where $t_i(c_i; \lambda)$ is the transmittance of the ith colorant layer, c_i is the concentration of the ith colorant (which varies between 0 and 1), and $d_i(\lambda) = d_i(1; \lambda)$ is the density at maximum concentration.

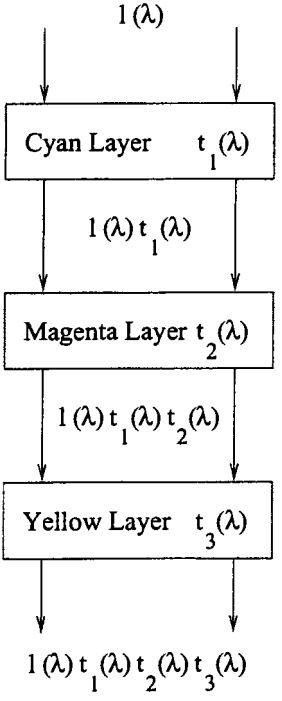


Figure 1.28 Subtractive color reproduction.

Using samples of the spectra involved, the spectrum of transmitted light can be represented as[300]

$$\mathbf{g} = \mathbf{L}[10^{-\mathbf{Dc}}] \tag{1.63}$$

where $\mathbf{L}$ = a diagonal matrix representing an illuminant spectrum
$\mathbf{c}$ = the vector of colorant concentrations
$\mathbf{D} = [\mathbf{d}_1\mathbf{d}_2\mathbf{d}_3]$

The remaining boldface symbols represent sample vectors of the corresponding spectral functions, and the exponentiation is computed component-wise.

For prints produced on paper, the transmitted light is reflected by the paper surface and travels once again through the colorant layers. This process is readily incorporated in the model of Equation 1.63 as an additional diagonal matrix that represents the reflectance spectrum of the substrate and a doubling of the densities $\{d_i(\lambda)\}_{i=1}^{3}$. Technically, for diffuse illumination the Kubelka–Munk model (see Reference 108, Chapter 7) should be used with the scattering terms set to zero. The mathematical details are, however, unaffected by this technicality, except for a change in the scaling of densities. For simplicity, the substrate reflectance can be conceptually included in the illuminant matrix $\mathbf{L}$, and the same equations can be used for reflective media. While the assumption of transparent layers with no scattering and no interaction between layers is sometimes too simplistic (for instance, for halftone prints and/or pigmented colorants), it is also fairly accurate for a number of useful cases including typical photographic slides and (to a lesser degree) photographic prints, and it has been successfully used in these scenarios.[261,262]

An example of a photographic contone subtractive color reproduction process is illustrated in Figures 1.29 and 1.30. The measurements used in these figures correspond to a Kodak IT8 photographic target.[217] Figure 1.29 illustrates the reflectances for white paper and for the maximum density cyan, magenta, and yellow colorants on the target. Figure 1.30 shows the estimated spectral densities for cyan, magenta, and yellow colorants (at maximum concentrations) as determined from these measurements. In these plots, the spectral regions corresponding to a high density value correspond to spectral wavelengths where the colorants strongly absorb light and regions of 0 density correspond to regions where the colorants transmit all the light through with no absorption. Note that the colorants all have relatively wide absorption bands with significant overlap between the absorption regions of the different colorants.

Even the simplified model of Equation 1.63 cannot be used for a closed-form calibration of a subtractive system. Analytical models therefore often assume that the three dyes have nonoverlapping rectangular-shaped absorption regions that cover the visible region of the spectrum. This is known as the *block-dye* assumption. Using the block dye assumption, colorant concen-

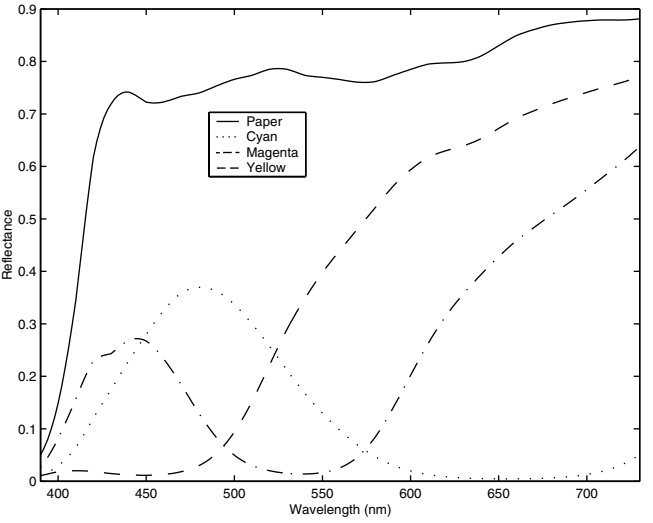


Figure 1.29 Spectral reflectances for white paper, cyan, magenta, and yellow from the Kodak IT8 target.

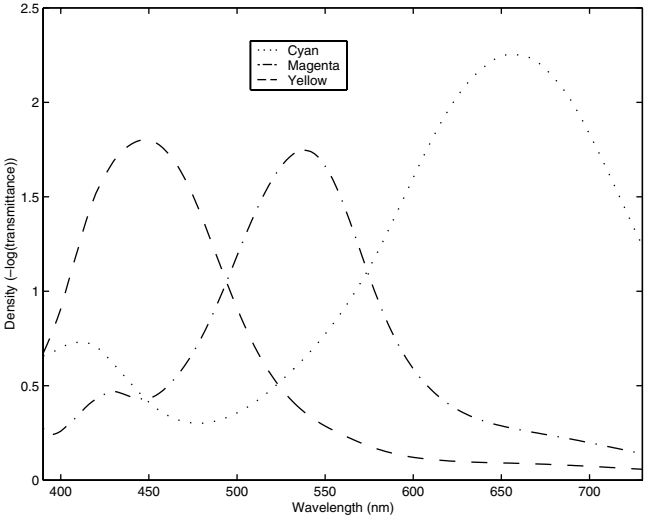


Figure 1.30 Spectral densities for cyan, magenta, and yellow from the Kodak IT8 target.

trations required to produce a given CIE tristimulus can be determined in closed form.[300] Under the assumption of nonoverlapping spectral absorption bands for the colorants, the colorants' interactions are eliminated, and the system can be transformed into an equivalent additive system. Because real

colorants deviate from this ideal desired behavior, the absorptions for a colorant in the visible band but outside its primary absorption band are often referred to as *unwanted absorptions.* Because unwanted absorptions are a hard reality of actual colorants, the utility of the block-dye assumption is restricted to educational illustrations of the subtractive process.

Most contone printers available currently use thermal dye diffusion technology. The coloring dyes in such a system are transferred from a dispensing film into the reproduction medium by means of heat-induced diffusion. Often, a fourth black dye is used in addition to the cyan, magenta, and yellow dyes to achieve greater contrast and better reproduction of achromatic (gray) colors. A review of the physical processes involved in a thermal dye diffusion printer can be found in References 113 and 114. Note that conventional photography uses subtractive color reproduction; thus, continuous tone printing is possible using photochemical methods that mimic photography. For an overview of this method and for more details of the dye diffusion printing process, the reader is referred to References 72 and 148.

In practice, the models for subtractive printers described above are applicable to only a small set of printers. For several applications, the colorants are not completely transparent; therefore, the Kubelka–Munk theory,[16,108,168] which accounts for scattering of light by the colorants, is more appropriate than the Bouguer–Beer law. Because accurate determination of the model parameters is fairly complicated, and there are interactions between the media and the colorant layers that are not accounted for even in the Kubelka–Munk theory, often purely empirical techniques are used to profile color printers. The model in Equation 1.63, though somewhat restrictive, has been successfully used to model and profile a thermal dye diffusion printer[17] and has also proven very useful in analytical simulations of printers and in making design choices.[139]

Typical empirical approaches for color printer calibration begin by measuring the color of test prints corresponding to a uniform grid of control values. This provides a sampling of the mapping from device control values to a device-independent color space. A variety of interpolation-based techniques are then utilized to determine the required inverse transformation, typically in the form of a lookup table over a uniform grid in a color space.[128,218,240] Interesting alternate approaches have also utilized neural networks[160] and an iterated scheme that concentrates measurements in regions of greatest impact.[40]

1.11.1.4 Halftone printers

Contone printers require reliable and accurate spatial control of colorant concentrations, which is difficult to achieve and control accurately. As a result, contone printers are rather expensive. Most desktop printers are therefore based on the simpler technique of halftoning, which has long been the color reproduction method of choice in commercial lithographic printing. Like CRT displays, halftoning exploits the spatial lowpass characteristics of the human

visual system. Color halftone images are produced by placing a large number of small, differently colored dots on paper. Due to the lowpass nature of the eye's spatial response, the effective spectrum seen by the eye is the average of the spectra over a small angular subtense. Different colors are produced by varying the relative areas of the differently colored dots. In contrast with contone printing, the concentration of a colorant within a dot is not varied, so halftone printers are considerably easier and less expensive to manufacture. Special processing of images is necessary to determine the dot patterns for the different colors prior to printing on a halftone printer. This processing is the subject of Chapter 6, "Digital Color Halftones," and only the halftone printing mechanism will be briefly discussed in this chapter.

For reasons similar to those for contone processes, most three-ink halftone systems use cyan, magenta, and yellow colorants for printing the dots (see Reference 341, Chapter 3). Just as in contone printers, a fourth black colorant is often introduced to conserve the more expensive colorants, reduce ink usage, and produce denser blacks (see Reference 341, p. 282).

A model for halftone printers was first suggested by Neugebauer,[213] who observed that the overlap of CMY colorant halftone dots produces eight primary colors: the paper (no-ink), each of the three colorants (cyan, magenta and yellow), combinations of two colorants (red, green, blue), and combinations of all three (black). The observation generalizes from the three-colorant case to the case of four or any arbitrary number of colorants. The colorants combine subtractively over the regions in which they overlap and, in general, produce up to 2^K different colors with K colorants. These distinct colors are called the *Neugebauer primaries*. Neugebauer proposed that the tristimulus values of the print can be expressed as the weighted average of the tristimuli of the Neugebauer primaries with the weights equal to their fractional area coverages. Due to the penetration and scattering of light in paper, known as the *Yule–Nielsen effect*† (or, alternatively, as the optical dot gain),[51,340] the simple Neugebauer model does not perform well in practice. As a result, several empirical modifications have been suggested for the model.[255] Recently, considerable success has been demonstrated in using a spectral version of the Neugebauer model with empirical corrections for the Yule–Nielsen effect,[9,11,252,308,336] in which the reflectance of a CMYK halftone print is modeled as

$$r(\lambda; \mathbf{w}) = \left(\sum_{i=1}^{16} w_i r_i^{1/n}(\lambda) \right)^n \tag{1.64}$$

where $r_i(\lambda)$ and w_i are, respectively, the spectral reflectance and the fractional area of the ith primary (there are 16 primaries for a 4-colorant [CMYK] system); and n is the empirical Yule–Nielsen correction factor. As a further

† Note that in the printing of the original paper,[340] Nielsen's name was misspelled as "Neilsen." Both spellings have therefore been used in existing literature.

simplification, the reflectances of the Neugebauer primaries composed of overprints of more than one colorant may be expressed in terms of the transmittances of the different colorant layers as was done in the subtractive model of Equation 1.63. However, because this assumption of additivity of densities reduces accuracy, it is usually not invoked; instead, the primary reflectances $\{r_i(\lambda)\}_{i=1}^{16}$ are directly measured from prints.

Normally, the fractional areas w_i are themselves related to the fractional areas c, m, y, k covered by the cyan, magenta, yellow, and black colorants, respectively. The relations determining the w_i from c, m, y, k depend on the method used to generate the halftone images. Equations for these relations were first derived by Demichel[69] by assuming independent random coverage for the colorants. Rotated halftone screens[341] that approximate this random independent coverage assumption are often used for printing, because they are robust to commonly occurring registration (alignment) errors between the different colorants images (*separations*). Alternative halftone screens have different characteristics and consequently different relationships between the fractional colorant coverages and the primary areas.[8,243]

Prior to the work of Neugebauer, halftone color reproduction was often confused with subtractive reproduction, and the Neugebauer model therefore offered very significant improvements.[242] However, the actual halftone process is considerably more complicated. As a result, several empirical modifications have been suggested for the model, such as the *Yule–Nielsen* parameter n and the spectral version of the model with the *Yule–Nielsen* modification. While these models have had some success, complete and accurate physical models for the color halftone printing process and the Yule–Nielsen effect continue to be elusive. Promising new directions are therefore a subject of continuing research.[74,75,111] Additional details on the research in this area can be found in Chapter 3.

One obstacle in the direct use of Neugebauer models in characterizing a halftone printer is that the relationship between the control values and the printing area of the different colorants is usually not known *a priori*. Hence, an empirical component is normally required even for characterization schemes using a model. This empirical component is in the form of one-dimensional pretransformations of device control values, which often serve the additional purpose of increasing characterization accuracy along the *achromatic* or *neutral gray* axis, where the eye has significantly greater sensitivity.[286] Purely empirical schemes similar to those used for characterizing contone printers can also be used for halftone printers. The models mentioned above are nonlinear and nonseparable in the device control values and cannot be inverted analytically. Hence, for both model-based and empirical schemes, the inversion of the characterization needs to be performed numerically. In either case, the final mapping from required color tristimuli to device control values is realized as a multidimensional lookup table. The models, however, have an advantage over a purely empirical approach in that they offer a significant reduction in the number of measurements required. An interesting generalization of the Neugebauer model is dis-

cussed in References 8, 9, and 252, where the model is interpreted as interpolating between a number of end points specified by the primaries. Accuracy is improved by using local interpolation over smaller cells, which in turn implies more measurements. The generalization, known as the cellular Neugebauer model, thus offers a graceful trade-off between accuracy and the number of measurements required. Due to the large number of effects ignored by most models, they can offer only limited accuracy. Therefore, in graphics arts and printing industries, where there is greater emphasis on quality, measurement-intensive empirical schemes are often preferred.[248] A comparison of some model-based and measurement-based empirical schemes for electronic imaging applications can be found in Reference 157.

Halftone printers have been manufactured using very different technologies for printing dots on paper.[148(p. 4–8)] The most promising current technologies utilize inkjet, thermal transfer, and electrophotography to produce the halftone dots. Even a brief mention of the principles and technology of these devices is beyond the scope of this chapter, and the interested reader is referred to References 72, 148, 241, and 254 for details.

1.11.1.5 Recent advances in color displays and printing

The increasing use of portable computers and hand-held mobile devices has fostered considerable research in a variety of display technologies. While the LCD display is dominant in this class, a number of alternative technologies are actively being developed. These include plasma displays, electroluminescent displays, and displays based on organic light-emitting diodes (OLEDs). A description of the historical development, physical principles, and relative merits and demerits of some of these technologies can be found in References 37, 155, 199, and 272. Most of them are additive color systems similar to a CRT and use a mosaic of red, green, and blue "dots" to produce color, although some LCD devices are based on the subtractive principle[155,321] or on spectrally selective reflection.[107]

A majority of the color display devices mentioned so far rely on the spatial lowpass characteristics of the human eye to produce different colors using a mosaic of differently colored regions. An alternative system for producing color, known as *field sequential color* (FSC), is based on the temporal lowpass nature of the eye's response. In an FSC system, red, green, and blue image frames are projected in rapid succession onto the viewing screen, and the temporal averaging in the observer's eye produces the illusion of a single colored image. An FSC system was originally selected by the FCC for color TV transmission but, before it could be commercialized, it was replaced by the monochrome-compatible NTSC system in use today. The primary drawback in such a system was the high frame rate and bandwidth requirements.[257(p. 218–219)] Recently, there has been a resurgence of interest in FSC systems. An interesting example of a recent FSC system is Texas Instruments' digital micromirror device (DMD)[339] that utilizes an array of deformable micromirrors. In the deformed state, each micromirror deflects light from an

illuminating lamp onto a single picture element (pixel) on the viewing screen. The duty cycles of the deformation of different mirrors are modulated to produce image intensity variations on the screen. Color is produced by placing a color filter wheel between the lamp and the micromirror device and synchronizing the red, green, and blue frames with the color wheel. Alternative configurations using three separate DMD devices or two devices in a five-primary projection system have also been reported.[338] From a color imaging standpoint, DMD displays are rather interesting, as they are almost linear and allow considerable flexibility in the choice of the primaries through the use of different color filters in the filter wheel.

There have also been significant new advances in color printing in recent years. Color halftone printers have continually improved in resolution, speed, and cost. Some devices now incorporate limited contone capability through a coarse variation in colorant concentrations/drop volumes. The gamut of printers has also been enlarged by using improved colorants or, more recently, by using more than three/four inks, which is referred to as high-fidelity (hi-fi) printing.[26,104,223] It is also common in inkjet printers to use dilute cyan and magenta colorants in addition to the regular cyan, magenta, and yellow colorants. This is particularly beneficial, because the lower density colorants produce more spatially uniform and less visible halftone structure in light regions such as Caucasian skin tones and pale blue skies, which are quite critical to the good reproduction of photographs.

The color profiling for several of the devices mentioned above is performed using model-based or empirical techniques already in use for existing devices. Other devices are still in their infancy, and little information if any is available on their color characterization and performance. As they find increased acceptance, there will no doubt be a greater demand for more accurate color characterization and for reasonable models of these devices. This, therefore, will be an active area of color imaging research in the future.

1.11.2 Image characteristics

The spectral radiance or reflectance of an image carries the most complete color information. Most color spectra are fairly smooth, and a 10-nm wavelength sampling is normally sufficient for representing these spectra.[303] The spectra of fluorescent gas-discharge lamps have sharp spectral peaks and therefore require a higher sampling rate or alternative model-based approaches.[189,264,302] A 10-nm sampling of spectra in the 400- to 700-nm region provides $N = 31$ samples† for each spectrum. Thus, color spectra lie in a 31-dimensional vector space. Color spectra are usually smooth, so they do not really exhibit 31 degrees of freedom. A principal-component analysis of common reflectance spectra reveals that a large fraction of the energy in

† The 400- to 700-nm interval is chosen for illustration and the exact interval is not central to the argument here. Typical color measurement instruments report spectra over this or a larger wavelength. region.

these spectra is contained in a few principal components.[149] Several researchers have investigated this problem and have estimated that color reflectance spectra can be represented by using anywhere between 4 and 12 principal components, depending on the accuracy desired.[54,194,314,319]

The spectral reflectance characteristics discussed in the last paragraph covered almost the entire range of naturally occurring reflective objects. Because most color reproduction systems exploit trichromacy and have only three additive/subtractive primaries, their spectra often have even fewer degrees of freedom. Thus, the spectra from a CRT display lie in a three-dimensional space defined by the spectra of the phosphor primaries. The same is true of all additive systems that utilize three primaries. For subtractive and hybrid systems, although there often are only three primaries, they are not confined to a three-dimensional linear space, due to the nonlinearities in the subtractive process. Despite the nonlinearities, the presence of only three degrees of freedom in the color production process may often be exploited in imaging applications.[270]

1.11.3 Computer-generated imager

The previous section focused on natural image scenes and *hard-copy* prints of these images. In the area of multimedia applications, the computer is an alternative, rapidly growing source of color images. Computer-generated scenes and animations are now common in video games and movies.[53,247] These applications strive for realistic images and therefore have to model the interactions between lights and objects in the scene, using both geometric optics and colorimetry to produce proper color, shading, and texture for the surfaces involved.[92(Chap. 14),93] For these applications to be successful in multimedia applications, it is necessary that the color information be recorded in a form that permits accurate rendering of the image on multiple output devices. This is particularly relevant for recording on movie film, which has entirely different characteristics from the CRT.[171]

Color can also be used to effectively encode information for presentation and visualization. Bar graphs and pie charts used in presentations are simple examples of the use of colors to distinguish different regions. The capabilities of color encoding are also exploited more fully in applications that use color to meaningfully represent quantitative differences in the data. Examples of such applications include visualization of multispectral/stereo data using *pseudo-color* images, and the use of color to portray three-dimensional relationships in multidimensional databases. The proper use of color in these and other multimedia applications greatly enhances their effectiveness.[92(p. 601),306] Usually, the color images are designed for viewing on a CRT monitor and tend to exploit the full capabilities of the monitor. When printing these images, it is necessary to preserve their information and effectiveness. This is often a significant challenge, because the printer gamuts are significantly smaller than monitor gamuts. Methods of compensating for this type of gamut mismatch are discussed in Chapter 10.

1.11.4 Color recording systems

Systems for recording color information include both color measurement instrumentation useful for large uniform regions and color image capture devices designed to capture spatially varying color information. Color measurement instruments are used primarily for determining the color characteristics of objects and of imaging input and output devices. They include both spectral measuring devices such as spectroradiometers/spectrophotometers and filter-based instruments that directly measure colorimetry. The correct color measurement instrument for a particular application depends on several factors including the cost of the instrument, the accuracy desired, the variety of viewing conditions for which a calibration is desired, and the number and type of devices that must be calibrated. Color image capture devices include conventional and digital color cameras, which record color information from a scene of real world objects, and color scanners, which record color information from a prior reproduction such as a printed color image, slide, or negative.

1.11.4.1 Spectroradiometers and spectrophotometers

Sampling of the spectral distribution provides the most direct and complete technique for recording color information. A *spectroradiometer* is a device that measures the power of optical radiation as a function of wavelength. Figure 1.31 shows the schematic cross section of a spectroradiometer. The light is collimated by the lens onto a dispersive element, which decomposes it into its spectrum. The spectrum is then sampled and recorded using single or multiple detectors. Typically, a diffraction grating is used as the dispersive element, because it provides an almost linear relationship between wavelength and displacement in the detector plane, as opposed to an optical prism, for which the correspondence is highly nonlinear. The linear relationship considerably simplifies calibration procedures.

Modern spectroradiometers use charge-coupled-device (CCD) arrays as the detectors because of their linear characteristics. A sampling of the spec-

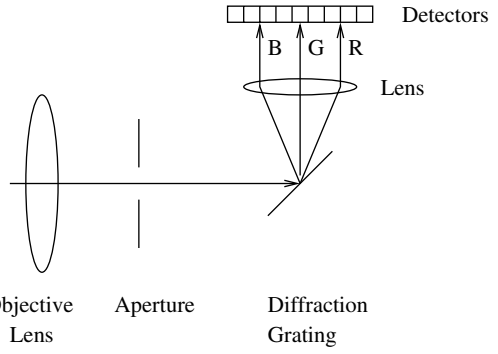


Figure 1.31 Schematic cross section of a typical spectroradiometer.

trum is achieved automatically through the placement of physically distinct detectors in the measurement plane. Because the separation between the detectors need not correspond directly to a convenient wavelength spacing, the detector outputs are usually interpolated to obtain the final spectral samples. Even though the CCDs are almost linear in their response at a given wavelength, their spectral sensitivity is not uniform. Therefore, a gain compensation procedure is usually necessary to obtain calibrated output from the device.[204(p. 338)]

The range and the sampling interval of spectroradiometers vary according to their intended application. Spectroradiometers used for color typically report measurements over the range of 380–780 nm and are capable of a spectral resolution ranging from 1 to 10 nm. For most color work (including that using fluorescent lamps as sources), 2-nm sampling is sufficient.[303] However, because some light sources have monochromatic emission lines in their spectra, a deconvolution of the spectroradiometer measurements may sometimes be necessary to obtain greater accuracy.[264]

Spectroradiometers can be used to measure both self-luminous and reflective objects. For the measurement of reflectance spectra, a spectrally smooth (preferably, white) light source and a spectrally referenced sample (preferably, white) are required. Comparison measurements between the known sample and the sample under question are made under identical conditions, allowing the determination of the unknown sample's reflectance spectrum, from which the color descriptor under any viewing illuminant can be obtained. Normally, the exact spectrum of the illuminant used with the spectroradiometer in the reflectance measurement is immaterial, provided it has sufficient power over the spectral range of interest. However, for the measurement of fluorescent materials, the power in the ultraviolet regions is also important, and the illuminant must closely approximate the desired standard illuminant under which colors are to be computed.[347]

While spectroradiometers can measure both luminous and nonluminous objects, they are expensive, larger, and less portable than the other instruments, and they generally are more difficult to operate due to their multiple operating modes. In particular, it can be difficult to set up a spectroradiometer for measuring reflectance samples under controlled conditions. There are automated systems for performing measurements on multiple samples using an x-y stage and a single light source.

An interesting application of spectroradiometry that extends beyond the visible spectrum is in multispectral scanners carried by remote sensing satellites. These scanners disperse radiation into different spectral bands in much the same way as the spectroradiometers discussed above. Early cameras in these satellites used 5 to 12 spectral bands extending from the visible into the infrared region.[36,287] The Airborne Visible Infrared Imaging Spectrometer (AVIRIS),[106] which samples the range of 400 to 2500 nm at 10-nm resolution, is an example of more recent scanners that use a larger number of bands.

A *spectrophotometer* is a device for measuring the spectral reflectance of an object. Unlike a spectroradiometer, a spectrophotometer does not measure

self-luminous objects. Therefore, spectrophotometers are useful for the calibration of printers and scanners but not of CRT displays. Spectrophotometers have their own internal light source that illuminates the sample under measurement. There are many geometries for the sensor/illuminant combination, and each may be optimal for different applications. For color imaging applications, the sensor and illuminant in the device are often set up for a 45°/0° condition as shown in Figure 1.32. Some devices illuminate the sample diffusely and measure off axis, allowing an option to include or exclude the specular component from the sample. Some devices contain an optical grating, others contain a filter wheel, and yet others use spectrally different light sources to illuminate the sample.

Spectrophotometers measure reflectance as the ratio of two (uncalibrated) spectroradiometric measurements[266, 347] using the same principle that was described in the last section. This is shown schematically in Figure 1.33. The light source is contained within the spectrophotometer and is used to illuminate both a standard sample with known reflectance, $r_s(\lambda)$, and the test object whose reflectance, $r_o(\lambda)$, is to be measured. If $l(\lambda)$ denotes the SPD of the light source, and the device makes K spectral measurements at $\Delta\lambda$ wavelength intervals in the region $[\lambda_0, \lambda_0 + (K-1)\Delta\lambda]$, the reference measurements can be represented as

$$m_s(k) = d_k l(\lambda_0 + k\Delta\lambda) r_s(\lambda_0 + k\Delta\lambda), \quad 0 \le k \le (K-1)$$

where d_k denotes the detector sensitivity at $(\lambda_0 + k\Delta\lambda)$. Similarly, the object measurements are given by

$$m_o(k) = d_k l(\lambda_0 + k\Delta\lambda) r_o(\lambda_0 + k\Delta\lambda), \quad 0 \le k \le (K-1)$$

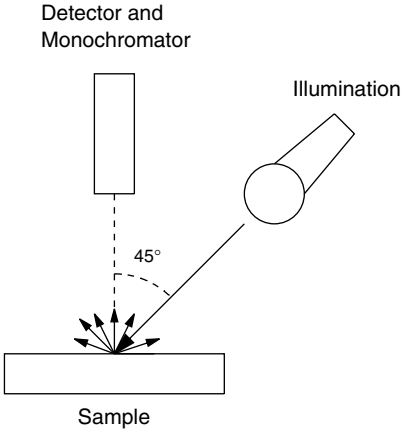


Figure 1.32 45–0° measurement geometry.

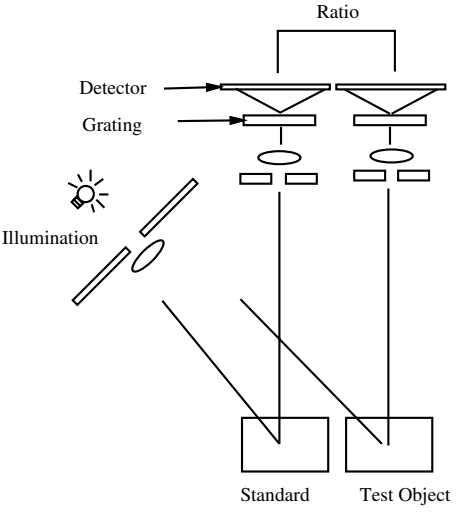


Figure 1.33 Spectrophotometer measurement.

The object reflectance samples can therefore be determined as

$$r_o(\lambda_0 + k\Delta\lambda) = \frac{m_o(k)}{m_s(k)} r_s(\lambda_0 + k\Delta\lambda), \quad 0 \le k \le (K-1)$$

Mathematically, it can be seen that the detector sensitivity, d_k, and the illuminant, $l(\lambda)$, cancel out and have no impact on the measurement. However, to obtain good performance in the presence of quantization and measurement noise and errors due to the limited dynamic range of the detectors, it is desirable that the product of these quantities be nearly constant as a function of wavelength. For similar reasons, it is desirable that the reflectance of the standard sample be close to unity at all wavelengths. To avoid unnecessary duplication of the optics and sensors, the measurements of the reference standard and the object are usually performed sequentially instead of using the parallel scheme shown in Figure 1.33. In addition, for added convenience and to save time, typical measurement devices make one measurement of the standard that is stored and used for a number of successive object measurements.

Because most real-world reflectances are relatively smooth functions of wavelength[303] and have low dimensionality as discussed in Section 1.11.2, most spectrophotometers work with much larger sampling intervals than spectroradiometers, typically reporting reflectance at 5-, 10-, or 20-nm intervals. The built-in illumination in these devices is usually a filtered incandescent or xenon arc lamp whose spectrum is smooth (unlike fluorescent lamps) and therefore does not unduly amplify the measurement noise and quantization errors. Spectrophotometers used in color work usually sample the spectrum in the 380- to 780-nm range, although the lower-wavelength end

of the spectrum may be truncated or less accurate in some devices because of the lower energy in incandescent lamps at the ultraviolet end. Owing to the lower resolution requirement, and because of the less stringent calibration required (due to the normalization of illuminant and detector sensitivities), spectrophotometers are considerably less expensive than spectroradiometers and are also more stable over time.

The design of spectroradiometers and spectrophotometers should account for a large number of factors excluded from the simplistic description given above. Both devices suffer from systematic and nonsystematic deviations from the ideal behavior described above and must be accurately calibrated to known radiant and reflectance standards prior to use. In particular, stray light, detector nonlinearity, effects of polarization, variations in illumination and measurement geometry, and unaccounted fluorescence and thermochromism of samples are sources of systematic errors. Detector noise and quantum fluctuations in photon flux are examples of random errors encountered in measurements. The reader is referred to References 108 (Chapter 8) and 109 (Chapter 9) for a thorough, though somewhat dated, account of the systematic errors in these devices and their calibration procedures. A more current, though brief, overview is also provided in References 16 (pp. 74–86) and 135 (Chapter 5). Detector noise models for older instruments that used thermal detectors and vacuum tubes are described in Reference 35, and a more recent account of noise models for semiconductor detectors of radiation is provided in References 28, 70, 144, and 204. Some methods for accounting and correcting some of the systematic errors in spectrophotometers are discussed in Reference 24. The propagation of spectrophotometric errors in colorimetry has also been analyzed in Reference 83.

Color recording devices that attempt to sample spectral information suffer from a number of obvious drawbacks. First, because the total energy in the spectrum is split into a number of spectral samples, a sizable measurement aperture and/or long integration time will be required for reliable measurements of the spectra. In addition, the required optical components make some of the spectral devices rather expensive and therefore inappropriate for desktop use. Finally, measurement devices that exploit trichromacy are less accurate but can offer acceptable color performance and significant speed advantage at a fraction of the cost. Spectroradiometers and spectrophotometers are therefore used primarily for color calibration, where the larger aperture and longer measurement times are not prohibitive (in contrast with devices for recording complete spatially varying images).

1.11.4.2 Colorimeters and photometers

As suggested by its name, a *colorimeter* measures color tristimuli and reports these as color values in CIE XYZ, CIELAB, or related color spaces. Some colorimeters have an internal light source for the measurement of color of reflective objects, whereas others measure only self-luminous or externally

illuminated objects. For some devices, tristimulus values for the sample under a few different illuminants are available. Most colorimeters are small hand-held devices with no moving parts and a single light detector. They achieve their spectral "separation" by way of color filters placed before the detector or with spectrally different light sources used for illumination.† To accurately report CIE colorimetry, colorimeters must record information from which the colorimetry can be derived. The sensitivities of these devices are therefore often designed to match a set of color-matching functions. It is common for filter-based colorimeters to use four independent filters, one each to approximate the $\bar{y}(\lambda)$, $\bar{z}(\lambda)$, and two to approximate the two individual humps of the $\bar{x}(\lambda)$ color matching functions.[16(p. 86)]

Colorimeters are less expensive than spectrophotometers and spectroradiometers, but they do not provide the detailed spectral information that allows the calibration of a printer for an arbitrary viewing illuminant. Those that measure self-luminous sources are used in the calibration of CRTs. *Photometers* are single-channel devices that provide a measurement of the luminance of a self-luminous or externally illuminated object. They are inexpensive and find use primarily in the calibration of CRTs when the chromaticity of the CRT phosphors is known.

1.11.4.3 Photographic film-based recording schemes

Photographic film is not a digital recording device; however, a brief discussion of this medium is worthwhile, as it often forms the primary input to many digital color imaging systems. Film used for color photography records the color information in three spectral bands corresponding roughly to the red, green, and blue regions of the spectrum.

The image to be recorded is focused by a lens onto the film. The film contains three emulsion layers with silver halide crystals that act as the light sensors and sensitizing dyes that make the crystals in the three layers respond to different spectral regions. Typically, the top layer is blue sensitive; this is followed by a yellow filter and green and red sensitive layers, respectively. The yellow filter keeps blue light from getting to the lower layers that are also sensitive to blue light. Light in each of the three spectral bands initiates the formation of development centers in the corresponding film layer. When the film is chemically processed, the silver halide crystals at the development centers are converted into grains of silver, and unexposed crystals are removed. The number of grains of silver in a given layer at a particular location is determined by the incident light energy in the image in the corresponding spectral band at that location. Thus, the spatial distribution of silver grains in the three layers forms a record of the spatial distribution of blue, green, and red energy in the image.

The relation between the density of silver grains and the incident light spectrum is highly nonlinear. In addition, the formation of silver grains is not deterministic, and the randomness in grain formation contributes to

† The latter configuration is disadvantaged in the measurement of fluorescent samples.[16(p. 75)]

noise in the recording process, known as *film grain noise*. Film grain noise is often modeled as a Poisson or Gaussian random process[143(pp. 619–622),258,259,294] and constitutes multiplicative noise in the recorded image intensity.[234(p. 342)]

An image record in the form of three layers of silver grains is obviously of limited use. Therefore, further chemical processing of the film is necessary. For the purposes of this discussion, it suffices to note that this processing replaces the silver grains in the red, green, and blue layers with cyan, magenta, and yellow dyes in accordance with the principles of *subtractive color reproduction*, which were be discussed in Section 1.11.1.3. A more complete description for color photography can be found in Reference 130, and simplified mathematical models for the process are described in Reference 234 (pp. 335–339).

As an aside, one may note that, prior to the invention of spectrophotometers and spectroradiometers, two techniques were developed to record the spectral information of entire images on (monochromatic) film. In the microdispersion method of color photography, the light from each small region of image was split into its spectral components using dispersive elements, and the corresponding spectra (of rather small spatial extent) were recorded on film. The second method, known as Lippman photography, recorded the color information in the form of a standing wave pattern by using a mercury coating on the rear of the film as a mirror. Both methods required extremely fine-grain film to achieve the high resolution required and long exposure times to compensate for the low energy at each spectral wavelength. The reader is referred to Reference 130 for a slightly more detailed account of these techniques.

1.11.4.4 Digital dolor cameras and scanners

Digital color cameras and color scanners are color recording devices that operate on similar principles, though their intended use is quite different and often poses different challenges. Both of these devices record color information by transmitting the image through a number of color filters having different spectral transmittances and sampling the resulting "colored" images using electronic sensors.

Digital color cameras are designed to capture color images of real-world objects in much the same way as conventional cameras, with the difference that the images are recorded electronically instead of using film. Because the scenes may involve moving objects, they typically have two-dimensional CCD arrays that capture the image in a single electronically controlled exposure. Different schemes may be used to achieve the spatial sampling and color filtering operations concurrently. One arrangement uses three CCD arrays with red, green, and blue color filters, respectively. In such an arrangement, precise mechanical and optical alignment is necessary to maintain correspondence between the images from the different channels. Often, the green channel is offset by one-half a pixel in the horizontal direction to increase bandwidth beyond that achievable by individual CCDs.[118] For econ-

omy and to avoid the problems of registering multiple images, another common arrangement uses a color filter mosaic that is overlaid on the CCD array during the semiconductor processing steps. Because the green region of the spectrum is perceptually more significant, such mosaics are laid out so as to have green, red, and blue recording pixels in the ratio 2:1:1 or 3:1:1.[226] Image restoration techniques are then used to reconstruct the full images for each of the channels.[30,32,59,293,301] Recently, a novel camera design has been developed that avoids the problems of both color filter arrays and of multiple images by using a single sensor and effectively layering the red, green, and blue filters by using the spectral selectivity of light penetration in silicon.[94] One of the aspects of color capture that is more challenging for color cameras than for scanners and colorimeters is the lack of control over scene illumination. While the eye adapts to the scene as described in Section 1.9.1, producing a visual appearance that is largely independent of the scene illumination, cameras do not incorporate these adapting mechanisms and consequently produce images with severe color casts/shifts. Algorithms for estimating the relevant scene illumination characteristics and correction of the casts are therefore necessary. Details on the image processing for digital color cameras can be found in Chapter 12.

Scanners are usually designed for scanning images reproduced on paper or transparencies and include their own sources of illumination. Because the objects are stationary, these devices do not need to capture the entire image in a single exposure. Typical drum or flatbed moving stage scanners use a single sensor per channel which is scanned across the image to provide spatial sampling. The single sensor makes the characterization of the device easier and more precise and also allows the use of more expensive and accurate sensors. For desktop scanners, speed is of greater importance, and they usually employ an array of three linear CCD sensors with red, green, and blue color filters. The linear sensors extend across one dimension of the scanned image. This allows three filtered channels of the image along a line to be acquired simultaneously. To sample the entire image, the linear array is moved optically or mechanically across the other dimension of the image. In another variation of these devices, three different lamps are used in conjunction with a single linear CCD array to obtain a three-band image from three successive measurements.

Colorimeters, digital cameras, and scanners can be mathematically represented by very similar models. In the remainder of this section, a scanner will be used for illustration of such a model. However, the same discussion applies to colorimeters and cameras with trivial modifications that will be pointed out where required.

The schematic of a typical desktop color scanner is shown in Figure 1.34. The scanner lamp illuminates the image, and the light reflected off a small area is imaged by the lens onto a beam splitter that splits the light into a number of channels with filters having different spectral transmittances (the typical case of three channels is shown in the figure). The filtered outputs are integrated over the electromagnetic spectrum by optical detectors to

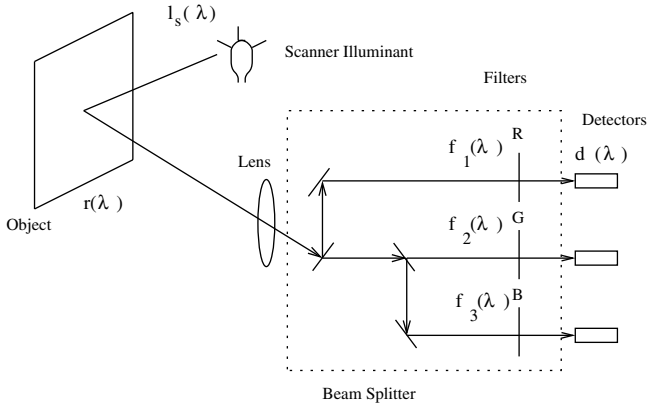


Figure 1.34 Schematic of a color scanner.

obtain a scanner measurement vector. This process is repeated over the entire image to obtain a "color" representation of the image. In actual scanners, the scanner measurements of the small area corresponding to a sampling unit are influenced by the color of the surrounding areas.[282] Ideally, restoration schemes should be used to remove the blur from the recorded image. However, due to the computational requirements, this is rarely done, and this aspect of the problem will be ignored in the subsequent discussion.

For sensors commonly used in electronic scanners, the response at a single spatial location can be modeled in a manner similar to Equation 1.1 as

$$\begin{aligned} t_i^s &= \int_{-\infty}^{\infty} f_i(\lambda) d(\lambda) r(\lambda) l_s(\lambda) d\lambda + \varepsilon_i \\ &= \int_{-\infty}^{\infty} m_i(\lambda) r(\lambda) l(\lambda) d\lambda + \varepsilon_i \end{aligned} \tag{1.65}$$

where K = number of scanner recording channels

$\{f_i(\lambda)\}_{i=1}^{K}$ = the spectral transmittances of the color filters

$d(\lambda)$ = sensitivity of the detector used in the measurements

$l_s(\lambda)$ = SPD of the illuminant

$r(\lambda)$ = spectral reflectance of the area being scanned

ε_i = measurement noise

$m_i(\lambda)$ = $f_i(\lambda)d(\lambda)$ is the product of filter transmittance and detector sensitivity

t_i^s = the value obtained from the ith channel

In a manner analogous to Equation 1.3, Equations 1.65 may be replaced by their discrete approximations using matrix vector notation as

$$\mathbf{t}_s = \mathbf{M}^T \mathbf{L}_s \mathbf{r} + \varepsilon \tag{1.66}$$

where $\mathbf{t}_s$ = the $K \times 1$ vector of scanner measurements
$\mathbf{r}$ = the $N \times 1$ vector of reflectance samples
$\mathbf{L}_s$ = an $N \times N$ diagonal matrix with samples of the radiant spectrum of the scanner illuminant along the diagonal
$\mathbf{M}$ = an $N \times K$ matrix whose ith column, $\mathbf{m}_i$, is the vector of samples of the product of the ith filter transmittance and the detector sensitivity
ε = the $K \times 1$ measurement noise vector

Note that, while these devices "sample" color spectra very coarsely, to assure that the above model is accurate it is necessary to meet sampling restrictions on the color spectra involved.[303] Due to their higher efficiency and lower heat dissipation, fluorescent lamps are often used in desktop scanners. Because their spectra have sharp spectral peaks, the sampling rate requirements (with uniform sampling) in the model of Equation 1.66 can be prohibitively high. A more efficient model for such a case is proposed in Reference 265, where a decomposition of the illuminant into the sum of a band-limited (smooth) component and impulses (monochromatic emission lines) is used to substantially reduce the dimensionality of the model while retaining the mathematical form of Equation 1.66.

For colorimeters and color cameras, the stimulus is normally a luminous object or an object illuminated by an illuminant external to the device. For these devices, the product, $\mathbf{L}_s\mathbf{r}$ (or its equivalent), defines the spectral radiance whose color is to be recorded. From the model in Equation 1.66, it can be inferred that, in the absence of noise, exact CIE XYZ tristimulus values can be obtained from the data recorded by colorimeters and color cameras if there exists a transformation that transforms the sensor response matrix, $\mathbf{M}$, into the matrix of CIE XYZ color matching functions, $\mathbf{A}$.[300] This is equivalent to the requirement that the HVSS be contained in the *sensor visual space* defined as the column space of $\mathbf{M}$.[310] For devices using three channels, this reduces to the requirement that $\mathbf{M}$ be a nonsingular linear transformation of $\mathbf{A}$. This fact has been known for some time and is referred to as the Luther–Ives condition.[131,184] Recent reiterations of this result can be found in References 102 and 127. A device that satisfies (generalizations of) the Luther–Ives condition will be said to be *colorimetric*.

For color scanners, the analysis is slightly more involved, because the illuminant used in the scanner is usually different from the illuminant under which the scanned object is viewed by an observer. Under these conditions, it can be shown that the CIE XYZ tristimulus values of the scanned object under the viewing illuminant can be determined exactly from the noiseless

scanner measurements if the human visual (viewing) illuminant space (HVISS) is contained in the *scanner visual space* (SVS) defined as the column space of $\mathbf{L}_s\mathbf{M}$. Because the spectra of fluorescent lamps used in most scanners is quite different from that of the daylight illuminants used in colorimetry, this condition is rarely met in practice. In addition, color tristimuli under multiple viewing illuminants often need to be estimated from a single scan of the image, and the above criterion would require an inordinately large number of detectors. In addition to the problems caused by fluorescent lamps, actual colorimeters, cameras, and scanners are subject to a wide variety of restrictions arising out of economic considerations and limitations of the processes and materials for manufacturing filters, optical components, and sensors. Techniques from signal processing are therefore useful for the evaluation and design of these devices.

It is the filters, $\{f_i(\lambda)\}_{i=1}^{3}$, over which the designer has the most control. A quality measure for evaluating single-color filters was first proposed by Neugebauer.[214] Recently, this was extended to provide a computationally simple measure of goodness for multiple filters in terms of the principal angles between the HVISS and the SVS.[310] The measure was used for the evaluation and design of color scanning filters.[311,312] The same measure was also successfully applied to the combinatorial problem of selecting an appropriate set of filters for a scanner from given off-the-shelf candidate filters.[313] A minimum-mean-squared-error approach, which requires more statistical information than purely subspace-based approaches, was introduced in Reference 316, where numerical approaches for minimizing errors in uniform color spaces were also considered. In Reference 317, noise was included in the analysis, and References 324–326 emphasize the reduction of perceived color errors in a hybrid device capable of measuring both reflective and emissive objects through the use of linearized versions of CIELAB space.[27] An alternate novel approach accounting for noise was proposed in Reference 76, where a filter-set was chosen from the multitude satisfying the Luther–Ives condition so as to minimize the perceptual impact of noise. In References 260 and 267, a unified treatment encompassing a number of these approaches is presented, and their performances are compared. Recently, the comprehensive figure of merit defined in References 260 and 267 has been extended through the inclusion of a signal dependent noise model and has been applied to the optimization of digital camera color sensitivities.[235–238]

It may be emphasized here that, for a rendition of the recorded reflectance scene under multiple viewing illuminants, more than three channels are usually necessary. A simple generalization of the Luther–Ives condition would require three K channels for K different viewing illuminants. In practice, however, between four and seven optimally designed spectral channels provide sufficient accuracy for common viewing illuminants.[268,317]

Note that, in recording color images digitally, both the spectrum and the spatial dimensions need to be "sampled." The different quality measures mentioned above consider only the spectral sampling aspect of the above problem. These are therefore suitable for evaluating the color recording fidel-

ity for large patches of uniform color and do not represent the complete performance for the image recording system. This is particularly true in CFA camera devices where the spatial and spectral sampling are performed jointly.

In actual devices, it is possible to have systematic deviations from the linear model of Equation 1.66. Sources of error include fluorescence of samples in scanners, stray light, inclusion of ultraviolet and infrared radiation in the measurements (which is not accounted for if the visible region of the spectrum is used in the model), and limited dynamic range of detectors.[86] However, if proper precautions are taken, these errors are small and can be included in the noise process with minimal loss of functionality.[86]

It should also be noted here that the above discussion applies to a system for recording color where the input spectra are not constrained to lie in a restricted set. In recording color information from color reproductions that exploit trichromacy and utilize three primaries, the requirements for obtaining precise color information are much less stringent, and sensors with any three linearly independent channels typically suffice. A proof of this result for a system using three additive primaries (whose spectra vary only in amplitude and not in spectral shape) appears in Reference 125. An example of an application where this can be readily seen is the measurement of colors produced on a cathode ray tube (CRT).[117] Note, however, that the calibration of these noncolorimetric recording systems is highly dependent on the primaries used in creating the images. Thus, they yield large color errors with images that are not produced with the primaries used in calibration.

For subtractive color reproduction systems (described in Section 1.11.1.3) that use varying densities of cyan, magenta, and yellow dyes to reproduce colors, one can conclude that any three sensors from whose measurements the densities can be inferred will suffice. In fact, in such a restricted system, not only is it possible to obtain colorimetric information about the original, but it is also possible to reconstruct complete spectral information from the three-channel record.[261,262,270] The mathematical characterization of this requirement requires assumptions on the spectra of the dyes and models for the specific processes used, which are discussed in References 261 and 270. In practical systems, it is sufficient to have any three reasonably narrow color filters with peaks in the red, green, and blue regions.[130(p. 247),249] Because this is far less demanding as a design objective than the colorimetric criteria discussed above, and because a large fraction of input images to scanners are in the form of photographic prints that use subtractive reproduction, most present-day scanners are designed to satisfy this requirement. The cost levied by this design trade-off is greater user intervention, because distinct calibrations of the scanner are required for accurately scanning reproductions produced with different subtractive primaries.[112] With the explosion in the number of color reproduction systems, it is difficult to maintain a correspondingly large number of calibration profiles and the color accuracy of such scanners is likely to be further compromised.

If scanners are designed to be colorimetric, a linear transformation, independent of the scanned object characteristics, can be used to accurately

estimate the CIE XYZ tristimulus values from the scanner measurements. However, due to the nonlinear relationship between density and tristimuli, scanners designed to measure dye densities perform poorly with a linear transformation. A number of heuristic nonlinear calibration schemes have therefore been used in practice. Three-dimensional lookup tables,[128] least-squares polynomial regression,[128,158] and neural networks[160] are examples of these approaches. Note, however, that these approaches offer significant gains over a simple linear transformation only when the characterization is performed for a restricted class of inputs.[112]

1.11.5 Multispectral recording and reproduction systems

A *multispectral* image is an image in which each pixel has *multi*ple channels that carry information about its *spectral* content. Multispectral images span the domain of images from conventional three-channel color images to hyperspectral imagery with hundreds of bands/channels used in remote sensing applications. Multispectral scanners often utilize narrowband spectral filters to record energy in different regions of the spectrum in a manner very similar to the color recording devices mentioned in Section 1.11.4.4. Traditionally, multiband sensors with more than three channels have been used in remote sensing applications. A major difference between these and the color recording devices arises from the fact that they are not attempting to capture information so as to satisfy a human observer. Therefore, these devices are not restricted to operating within the visible region of the electromagnetic spectrum and typically use infrared, visible, and microwave regions of the spectrum.[36] For the same reason, while dimensionality reduction of recorded data is often done while processing (see Section 1.11.5.1), there is no direct analog of trichromacy in remote sensing.

Traditionally, color imaging has worked with three channels, both for capture and reproduction.† Recent years have, however, seen a significant interest in multispectral image capture in the visible range of the spectrum and reproduction of multispectral images using objectives beyond those of simple color matching.

A variety of schemes have been proposed for multispectral image capture in the visible range. Devices in existence today are experimental, and most of these perform image capture using a single-channel camera in front of which are placed a number of spectrally selective filters that provide the separation of the image into multiple bands. Proposed filtering options include conventional absorption filters,[297] electro-optic tunable filters,[115] and interference filters[121] that offer narrow bandwidths. While absorption filters produce wider spectral bands than the electro-optic tunable filters and the interference filters, they also exhibit significantly less variation in their spectral transmittance with change in angular placement, which is often a source

† While the use of four colorants is common in color printing, and additional colorants have also been used in hi-fi color reproduction, the traditional approaches have chosen these in a constrained fashion, with essentially only three independent channels.[270,271]

of error in these systems. Because the number of channels is large, color-filter array type schemes that jointly sample the image spectrally and spatially are not common, though the use of filters in conjunction with trichromatic cameras has been investigated.[138] The large number of filters also implies a longer time for the scene capture and, consequently, the current devices are capable of capturing only static scenes with no motion, such as paintings or still-life scenes. For document imaging applications, the original to be captured is often a reproduction, and in these cases it is often feasible to recover spectral information from a simple three-channel record and realize the benefits of multispectral imaging.[20,262,270]

Multispectral capture offers several benefits over conventional three-channel capture. All of these advantages stem from the fact that the additional spectral data represent the information in the original scene more completely. The data can be used, for instance, to simulate changes of the SPD of the scene illuminant,[116] which may then be used for display or print renderings of the original as it would have appeared under a desired illuminant.† The use of multispectral data for this purpose is similar to the use of spectrophotometers in color characterization, where colorimetry is then calculated from the spectral measurements by specifying a desired illuminant.‡ The additional information in the multispectral capture also enables analysis of characteristics of the original[22,193] that would not be captured by conventional color capture. This is particularly valuable in the digitization of art archives and provides a mechanism for preserving these digitally, allowing better restoration and the ability to share them as more realistic reproductions. Several academic and government institutions have, therefore, initiated research in this area.

Additional capabilities of captured multispectral data are realized in conjunction with multispectral output systems that use more than three channels for output and can incorporate additional criteria for reproduction. The additional capabilities enabled by multichannel output devices can be used to try to effect a spectral match between the reproduction and the original that was (multispectrally) recorded. Such a reproduction would eliminate illuminant metamerism and provide a color match between the original and the reproduction across any viewing illuminant. The complicated nature of color hardcopy reproduction and limited availability and capability of colorant materials make this ideal goal extremely difficult to realize.[18,305] Some preliminary experimental results on spectral hardcopy reproduction using six colorants have recently been reported.[292] The large

† As already noted in Section 1.9.2, a color appearance model is not designed to address this problem. However, from the arguments at the end of Section 1.9.2, it is also clear that the benefit of multispectral capture for this application can be evaluated by comparing its efficacy over the incorrect use of a color appearance model in this situation. Such an evaluation is offered in Reference 116 using CIELAB as a crude appearance model.

‡ Note also that this application suffers from the same pitfalls as the color measurements in that the effects of fluorescence, if any, can be comprehended only to the extent that the illuminant being simulated is similar to the illuminant under which the scene is captured.

dimensionality of the spectral space ensures that, for any reasonable number of colorants, most spectra are outside the spectral gamut of the reproduction system. Practical systems in the near future can only hope to provide approximate spectral matching at best. One limitation with this approach is that the differences between the spectra of the original and the reproduction might be visible under common lighting conditions, and, even under a single lighting condition, color differences could result that could be eliminated by colorimetric matching. Thus, it is beneficial to consider alternative reproduction objectives for multispectral output. One straightforward extension is to minimize color differences under multiple common illuminants. This addresses illuminant metamerism for the standard observer but does not address observer metamerism arising due to differences among observers. An alternative objective is to consider the color variations among observers and use the freedom offered by the additional channels to minimize the impact of observer metamerism.[121] This is particularly meaningful for additive multiprimary displays and has been recently applied in that context.[121,337]

In addition to their use in color imaging applications outlined above, multispectral data are also extremely useful as a research tool for the realistic simulation of conventional imaging systems.[31,309] Such simulation and analysis can help direct improved design of these recording systems. The encoding of multispectral images is thus far in experimental formats, but some recent work has proposed mechanisms for exchange and an encoding format that is compatible with colorimetry in that three of the channels are selected to correspond to colorimetry.[165]

1.11.5.1 Principal-component recording

The color recording devices of Section 1.11.4.4 attempt to sample the spectra of images while preserving visual information. A recording of the spectrum itself provides greater information but is extremely slow and expensive. Because spectral information of reflective images is extremely useful for determining color under different illuminants, alternative schemes for recording their spectral information are of interest.

Note that, in the absence of noise, the scanned image in Equation 1.66 can be directly used to determine the projection of the image spectra onto the SVS. Hence, to obtain good reconstruction of reflectance spectra, the sensors can be chosen so that a large fraction of the energy in reflectance spectra lies in the SVS. In the absence of noise, the Karhunen–Loève (KL) transformation provides the mathematical solution to this problem in terms of the statistics of the ensemble of reflectance spectra.[126] The best spectral reconstruction of scanned spectra in a mean-squared error sense is obtained from a K channel scanner when the SVS corresponds to the span of the K *principal components* of the reflectance spectra, i.e., the eigenvectors associated with the K largest eigenvalues of the spectral reflectance correlation matrix.

The reflectance spectra of most naturally occurring objects are smooth functions of wavelength; the same is true of spectra produced using pho-

tography, printing, or paints. As a result, these spectra can be accurately represented by a few principal components. Various studies of reflectance spectra have estimated that between three and seven principal components (depending on application) provide satisfactory reconstruction of reflectance spectra for most color work.[54,122,198,314,319] Note that this offers a significant reduction in dimensionality in comparison with spectrophotometric measurements using uniform sampling.

Linear models for object reflectance spectra based on the principal-components idea have been used by many researchers for recovering illuminant and surface reflectance data from recorded images and for color correction applications.[41,122,315,319] Most of this research used KL transform on a spectrophotometrically recorded ensemble of reflectance spectra, and the problem of designing spectral recording devices based inherently on the principal-components approach has received little attention. There are, however, commercial color measuring devices that attempt to reconstruct spectral data from sensor measurements.[42] In addition, the principal-components approach has been used in analyzing multispectral satellite imagery, and the idea of a recorder based on principal components has also been suggested for acquiring satellite images.[287(Chap. 7)]

One may note here that some naturally occurring reflectance spectra do not adhere to the smoothness assumption. Examples of such spectra are colors produced due to multiple film interference in certain minerals and iridescent colors on some bird feathers and in shells containing calcium carbonate.[210(p. 261),267] A principal-components scheme leads to relatively large errors in such spectra. Hence, in imaging applications involving these objects, the principal-components approach would be inappropriate for approximation of their spectra.

1.11.6 Quantization and coding

Color images recorded with the different input devices described in the last section need to be quantized for digital processing. Both scalar and vector quantization techniques can be used in the quantization of color data. For simplicity, most color devices do independent quantization of the RGB channels with 8 to 12 bits per channel, with either uniform quantizers or companded quantizers that perform a gamma correction before the quantization. As mentioned earlier, the gamma correction significantly reduces the perceptibility of quantization errors, particularly in the eight-bit devices. For computer color displays based on a frame buffer architecture,[298] often only 8, 12, or 16 bits of video memory are allocated to each pixel, thereby allowing simultaneous display of only 2^8, 2^{12}, or 2^{16} colors, respectively. Vector quantization techniques have therefore been used extensively for displaying images on these devices. Chapter 9 surveys and discusses these techniques.

With the proliferation of digital color imagery, the problem of coding color images for transmission and storage has gained increased importance.

It was recognized early on that the highly correlated RGB spaces were not suitable for independent coding.[233] Consequently, most of the methods transform the data into a luminance channel and two chrominance channels that are then coded independently. A luminance-chrominance space also allows coding schemes to exploit the properties of human vision by allocating significantly fewer bits to the high-frequency chrominance components, which are perceptually less significant.

The most prevalent compression scheme at present is the JPEG standard for still images[228] and the MPEG standard for video data.[207] These standards are both based on the discrete-cosine transform (DCT).[5] While these standards do not explicitly specify the color spaces to be used, in current implementations, it is common to use the YCrCb space,[39] with the Cr and Cb components subsampled by a factor of two along both spatial dimensions.[212] The YCrCb color space is a luminance-chrominance color space, based on gamma-corrected RGB, that has been proposed for use as a standard in HDTV. The Y component is a luminance channel similar to L^*, and the Cr and Cb are opponent chrominance channels similar to a^* and b^*, respectively. The chapter on color image compression provides more details on the compression of color images and the new JPEG-2000 standard.

1.11.7 Device color spaces

The measurements from color recording devices and the control values for color output devices are the color representations of recorded images or the images to be reproduced, respectively. Hence, it is common to say that these values represent colors in the device's color space. Thus, typically there are RGB color spaces for scanners, cameras, and other input devices, and there are CMY/CMYK color spaces for color printers. Unlike the CIE standard color spaces discussed in Section 1.5.1, most of these color spaces are not standard and cannot be directly used for the meaningful archival/communication of image data. However, if these device spaces are related to the standard color spaces in a clear unambiguous way, these can also be potentially used for the exchange of color information. An advantage of such a scheme is that no transformations of the image data are required for display if the data are specified in the native color space of the device. Because a large fraction of the images on the World Wide Web are primarily targeted for display on CRT monitors that have very similar characteristics, a new standard color space, sRGB, has been proposed for use based on these characteristics.[6] The sRGB color space is basically a gamma-corrected tristimulus space that uses the CRT phosphors as primaries for determining the CMFs. In addition, the sRGB standard includes provisions for specifying the viewing conditions (white-point chromaticities, image surround, flare, etc.). The sRGB space is tied to the characteristics of common CRT displays and is therefore limited in some respects. In particular, there are colors that can be produced on common printers but lie outside the range of colors that are can be encoded in sRGB. These colors tend to be in the cyan and bright

yellow regions. Recently, extensions to the space to remedy some of its limitations with regard to gamut have been proposed.[283] However, as displays evolve to newer technologies, the original motivation for using the sRGB standard as a method for keeping data in a format that requires minimal transformation is also no longer valid.

Standardized CMYK spaces have also been defined in the graphic arts industry to allow color data to be supplied to printing press operators in a form that can be unambiguously interpreted (unlike device CMYK specific to a single press). The standards in this area vary by geography and include SWOP[288] (North America), Euroscale (Europe), and Japan Color (Japan). These standards often combine multiple independent specifications based on the printing conditions (e.g., for coated vs. uncoated substrates).

1.12 Color management and calibration

For proper color reproduction, the input and output devices involved must be calibrated. Historically, the systems used for color reproduction were calibrated† in a closed-loop configuration. As shown in Figure 1.35, in a closed-loop configuration, the complete system is calibrated from input through output. Thus, for color photography, the film sensitivities, dye absorptances, and developmental interactions were appropriately chosen so as to result in acceptable reproduction. In offset printing, the scanner was used to generate CMYK "separations" that were suitable for generating

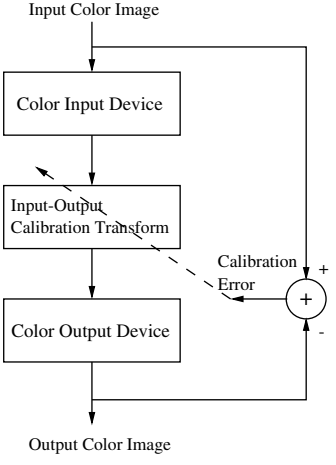


Figure 1.35 Closed-loop system calibration.

† Note that, in Chapter 5, the terms *calibration* and *characterization* have specific meanings as per-channel corrections and the full three-dimensional color correction that apply to the "calibrated" device. In this chapter, we do not follow that convention strictly, because the terms *calibration* and *characterization* are in common use with overloaded meanings. The use of these terms in this chapter should be apparent from the context.

halftone prints. With the increased use of digital computers and the evolution of desktop printing, it became obvious that such an approach has severe limitations. In particular, as the number of devices increases, calibrations for each input–output device pair are difficult to construct and maintain. In addition, because the calibrated data in a closed-loop calibration scheme are specific to one output device, they are not suitable for archival purposes or exchange with devices outside the system.

With the growth of networking and increased exchange of color image data between geographically divided systems, it was recognized that several of these problems can be solved by calibrating each device to a standard device-independent (DVI) color space, which can then be used for the exchange of data between different devices and for archival uses. As shown in Figure 1.36, in these systems, the data from an input device are converted to a device-independent color space and then transformed into the device-space of the target device for reproduction.

To enable proper management of color, several components are required. Color measurement instrumentation discussed earlier in Sections 1.11.4.1 and 1.11.4.2 is necessary for calibrating input and output systems. Standard formats for the storage and communication of these device calibrations are required so that different applications can make use of the calibrations. Also necessary are systems and algorithms that use the calibrations effectively to achieve desired results. These components are briefly outlined in the remainder of this section. As with other topics, several are discussed in detail in subsequent chapters.

1.12.1 Calibration and profiles

Calibration of a color imaging device relates its input/output to DVI color values. For an input device, calibration provides a mapping from device measurement values (e.g., scanner RGB) to DVI color descriptors (e.g., CIE XYZ, CIELAB etc.), and, for an output device, the calibration process yields a mapping from DVI color descriptors to device control values (e.g., CMYK, monitor RGB) which produce those color descriptors.

1.12.1.1 Input device calibration

To calibrate a scanner, the first step is to select a collection of color patches that span the gamut of interest. Ideally, these colors should not be *metameric* for the scanner or the eye (under the illuminant for which the calibration is being produced). Metamerism is defined as the property in which different spectra map to the same values under a set of sensitivity functions. These patches are measured using a color measurement instrument, such as a spectrophotometer or a colorimeter, which will provide the device-independent color values $\{\mathbf{t}_k\}_{k=1}^{M_q}$, where M_q denotes the number of patches. Any of the (device-independent) color spaces such as CIE XYZ, CIELAB, etc. can be used for this purpose. The use of CIELAB is common, as this space includes

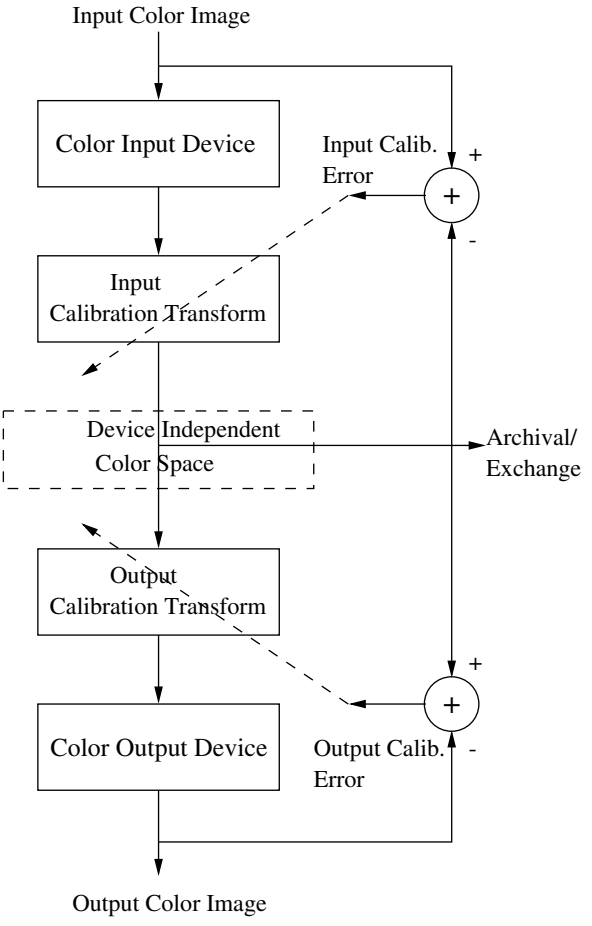


Figure 1.36 Device-independent color calibration.

information on the viewing illuminant in the white point. The patches are also measured with the scanner to obtain the scanner measurements $\{\mathbf{u}_k\}_{k=1}^{M_q}$.

To determine the CIE values for an arbitrary measured patch, the collected data are used to construct an interpolating function that maps from the space of scanner measurement values to the chosen device-independent color space. This function, $F(\cdot)$, can then be used to relate any of the scanner RGB values to colorimetric XYZ values or LAB values, i.e., $F(\mathbf{u}) = \mathbf{t}$. Normally, a parametric form is chosen for $F(\cdot)$, and the parameters are determined through regression. Several different schemes have been used for this pur-

pose, ranging from straightforward linear and polynomial regression to neural networks.[159(Chap. 11)] Usually, $F(\cdot)$ is complicated and computationally expensive. For this reason, $F(\cdot)$ is usually used to produce a finely sampled lookup table (LUT) from which $F(\cdot)$ at arbitrary points is obtained by using simple interpolation schemes.[159(Chap. 4)]

Calibration for digital cameras and video cameras is usually done in a similar fashion by using a target of patches with known reflectances or color values. Because the eye is very sensitive to deviations from the neutral (achromatic) colors, sometimes an additional one-dimensional transform is included on each of the RGB channels so that the $R = G = B$ line corresponds to neutral colors. This procedure is commonly referred to as *gray/white balancing*.

1.12.1.2 Output device calibration

For calibrating an output device, a transformation from DVI color values to the space of device control values is required. This requires a two-step procedure. In the first step, the printer characterization, which determines the forward transformation from printer control values to DVI color values, is determined. Then, this forward transform is used in the next step to determine the inverse mapping from DVI color values to device control values.

Because CRT monitors are represented well by the parametric models described in Section 1.11.1.1, the forward characterization of these devices is usually done by determining the model parameters from a few measurements. It can also be readily seen that, due to their additive nature, the inverse transformation from CIE XYZ (or other tristimulus) values to the CRT control voltages can be computed by means of a simple matrix transformation followed by a one-dimensional transform for the gamma correction. This scheme is used in all cases except those requiring the highest accuracy, for which LUT-based schemes may be used, and additional corrections may be made for the surface reflection (flare) from the monitor screen. In addition, it may be necessary to correct for the significant spatial nonuniformity over the CRT screen.[63,266]

For the forward characterization of printers, an empirical scheme similar to that described for scanner calibration is commonly used. By selecting a set of printer control values $\{\mathbf{c}_k\}_{k=1}^{M_p}$ covering the range of allowable control values, measuring the corresponding DVI color values $\{\mathbf{v}_k\}_{k=1}^{M_p}$, and using some interpolation scheme, the forward mapping, $\mathbf{v} = G(\mathbf{c})$ from control values to DVI color values is determined. For halftone printers, alternatively, the Neugebauer models mentioned in Section 1.11.1.4 have also been used. Because even the models are nonlinear and not readily invertible, for the inverse mapping $G^{-1}(\cdot)$, an interpolating function invariably is used to get a finely sampled LUT. Due to the four degrees of freedom in the control values of CMYK printers, there exist multiple control values that result in the same printed color. Because this poses a challenge in obtaining a smooth inverse mapping, the inverse often determines the

amounts of three virtual CMY colorants from which the CMYK control values are obtained by some functional relation. This process is viewed as the incorporation of black ink and removal of underlying CMY inks and is commonly referred to as *undercolor removal* (UCR). Typically, the UCR mapping from the virtual CMY values to actual CMYK control values is designed so as to better render achromatic colors, reduce total colorant amounts (for faster drying/better adhesion to paper), and (in some cases) reduce the use of the expensive CMY colorants.

1.12.1.3 Device profiles

To make the calibration transformations available to different applications that wish to use them, the calibration transformation for each device is stored in a device profile. In the early days of color management, different manufacturers used their own proprietary formats for the storage of these profiles, which were therefore useful only for applications from the same manufacturer. To realize the full benefits from the DVI calibration of devices, the desktop publishing industry is increasingly moving toward open systems. A standard format for the storage of device profiles has been defined and is being widely adopted.[62] This International Color Consortium (ICC) profile format specifies a wide variety of input and output device profiles suitable for efficiently representing the color calibration information. The use of a standardized format allows the profiles to be used by different applications from different manufacturers. Limitations of the profiles in some respects have resulted in some nonstandard implementations that have compromised compatibility, but these are being addressed by the ICC with input from member corporations.

1.12.2 Color management systems

A color management system (CMS) is responsible for interpreting the device profiles and performing the appropriate transformations to and from the device-independent space. The goal of a CMS is to provide predictable and consistent color without requiring specialized skills from the user. Thus, CMSs tie together device profiles with *color matching modules* that use these device profiles to transform device-dependent image data to DVI color spaces or to the device color spaces of target output devices on which the images are to be displayed. In addition, the CMS provides the user with flexibility to choose different *rendering intents* for different images. Thus, for instance, in reproducing a company logo, a perfect colorimetric match is usually desired; in producing bar graphs and pie charts for presentation, it is desirable that the colors be highly saturated for maximal impact; and in reproducing pictorial images, it is desirable that the closest perceptual match be obtained (which will depend on viewing conditions). Color management functions can be performed at several different phases of the imaging process — in the devices (e.g., Adobe's Postscript level 2/3 for

printers), in device drivers (e.g., Cannon Colorgear), applications (e.g., Adobe's Photoshop/Acrobat), or in the operating system (e.g., Apple's ColorSync/Microsoft Windows 2000/XP). A description of some of the practical aspects of color management in printing applications and a listing of vendors of color management software and solutions can be found in Reference 137. The notion of embedding color management in the operating system has the potential of making the process transparent to the end user. The system, however, also introduces limitations that can result in unexpected behavior when users are unaware of the color management or have incompatible components. As a result, even though several vendors of operating systems for desktop and workstation computers have incorporated CMSs into their products, these are not extensively utilized at present.

Ideally, with color management, one could accurately transfer color information from one medium (e.g., a CRT) to another (e.g., print). Unfortunately, this is an extremely difficult task for two reasons:

1. There are significant differences in gamuts of different devices (this was mentioned and demonstrated in Section 1.11.1).
2. The difference in typical viewing conditions for different media implies that a simple colorimetric match does not give an appearance match.

There is therefore significant interest in *gamut mapping* algorithms that map the colors in an image to suitable colors that can be reproduced on the target device. Methods that model the adaptations of the eye and allow the computation of appearance matched images under different viewing conditions are also an active area of research.

1.12.3 Gamut mapping

Gamut mapping is the process of mapping the displayable colors from one media to those of another media. As defined earlier, the ideal goal for color matching depends on the type of image and the intent of the user. The problem is probably most acute for pictorial images for which, ideally, one would like to map the colors so as to obtain the best possible appearance match between the images on the different media.[206,286] The strategy for gamut mapping could be either image dependent or image independent. Because image-dependent methods can use different strategies for different images, they often produce better results. However, they are also significantly slower than image-independent techniques, because they require a fresh computation for each image and are therefore seldom used in automatic gamut-mapping schemes. Several color applications, however, indicate out-of-gamut colors in an image via a key color or a flashing highlight (on a CRT). This allows the user to perform a transformation such that these colors are mapped satisfactorily. Such an approach, however, requires considerable operator skill to obtain satisfactory results.

The simplest technique of gamut mapping (particularly for CRT monitors) is clipping in the space of device control values. Because the space of control values is not a UCS, clipping of control values does not yield the closest printable color (as perceived by an observer). To remedy this problem, a simple extension would be to map out-of-gamut colors to the nearest in-gamut color in a UCS. While this approach offers significantly better results than the device space clipping,[98] it can often result in unacceptable hue shifts that are perceptually very objectionable. An additional limitation of this and the clipping approach is that smoothly varying regions beyond the device gamut can potentially be mapped to a single color, creating undesirable abrupt edges in the previously smooth regions and causing loss of significant information (such as shape from shading).

Another approach used for gamut mapping is a gamut compression algorithm that compresses all the colors in the image in a manner that reduces the colorimetric dynamic range in the image while ensuring that the colors can be reproduced. For example, one could move all the colors in the image toward one point, such as a mid-gray, until all the colors in the image are within the device gamut. Unlike the clipping approach, this method will retain some of the variation in smoothly varying image regions that are beyond the device gamut. In addition to UCSs, notions of hue, chroma, and saturation/value are extremely useful in gamut mapping research, as these can be directly related to viewers' objections to artifacts produced by gamut mapping algorithms and can be used in appropriately choosing the clipping and compression. In Section 1.7, it was pointed out that, in addition to being a UCS, the CIELAB space allows the computation of correlates of lightness, hue, and chroma. As a result, CIELAB has been used extensively in gamut mapping research. In the process, some limitations of CIELAB have also been discovered and have been addressed through empirical modifications. The "blue hue nonlinearity" was one such limitation that was discussed in Section 1.7.3.

The transformations to and from the perceptual spaces are usually nonlinear, and speed is often an issue.[60] For this reason, some CMSs may use a high-resolution LUT along with a linear interpolator to perform the transformation. Performing multiple transforms on the data can result in a loss of fidelity and introduce visually noticeable errors due to the accumulation of errors associated with finite precision arithmetic. Some CMSs can cache input and output transformations, concatenate them, and perform these in a single operation when the final image is desired. This not only saves time but also improves accuracy. Some of these powerful techniques are discussed in Chapter 11.

1.12.4 Appearance matching

Section 1.9 briefly discussed the adaptation in the eye in response to change in viewing conditions. Since different color reproduction media have different viewing conditions, in reproducing images across these different media, an appearance match instead of a pure colorimetric match is usually desired.

The simplest instance of appearance matching is the *white-point* matching method based on the von Kries transformation as mentioned in Section 1.7. This transformation converts tristimuli into a space of cone responses and applies a diagonal correction matrix that equates the white points under the two viewing conditions.[335(p. 432)] White-point matching is often used in color imaging applications, and there is support for it in several color management applications and packages such as PostScript.

In addition to white-point adaptation, there are several well characterized psychophysical effects that change with change in viewing conditions. In particular, it is well documented that the apparent contrast (perceived intensity gradient) of an image decreases in a dark surround in comparison to a bright surround.[134(pp. 49–57)] Often, this change in contrast is modeled by relating the luminance to the perceived lightness as a power–law relation, similar to that for CIELAB in Equation 1.32, with the exponent increasing as the surround gets brighter.[80,134(pp. 56–57)] Thus, *gamma correction* has also been used extensively to compensate for these effects, in addition to correction for monitor nonlinearity.

There has been considerable research in defining color appearance models that account for chromatic adaptation, influence of surround, and other psychophysical phenomena that affect the perception of images.[81] These models hold tremendous potential for use in cross-media color reproduction. The CIE is involved in an ongoing attempt to define a standard appearance model for use in imaging applications. An interim version of the model is available as a standard.[50] The model has also been recently refined in References 82 and 176. More details on the status of color appearance research can be found in Chapter 2, which is devoted to this topic.

1.13 Summary

This chapter provides an introduction to the fundamentals of color science and technology as applied to color digital imaging. It also attempts to provide a systems view of color imaging systems, where the interactions between image capture devices, image and color processing operations, image display and printing systems, inherent characteristics of natural and man-made images, and the human observer are outlined. Given the breadth of subject matter covered in the chapter, the presentation of several aspects is quite terse and abbreviated. A much more detailed view of several of the individual "components" within this system is presented in the remaining chapters of this book.

Acknowledgments

Parts of the material included in this chapter have been published in survey/tutorial papers I have co-authored.[266,269] I would like to thank my co-authors on those papers for their contributions. Several colleagues have also provided data and help with generating some of the figures included in this

chapter and I am grateful to them for their assistance. Finally, the insights (if any!) in this chapter reflect the knowledge I have gained from numerous colleagues, mentors, instructors, researchers, and authors who have helped shape my understanding of color science, color perception, and color imaging. I am grateful for having the opportunity to work in this wonderful field and to interact with you. Any errors/inaccuracies in the presentation can be attributed solely to my own lack of understanding.

References

1. ASTME 1360, *Standard Practice for Specifying Colors by Using the Optical Society of America Uniform Color Scales System,* American Society for Testing and Materials, West Conshohocken, PA.
2. AATC, CMC: Calculation of small color differences for acceptability, AATC Test Method 173–1992, in *AATC Technical Manual,* 1994.
3. E. Q. Adams, A theory of color vision, *Psychol. Rev.,* 36, 56–76, 1923.
4. E. Q. Adams, X-Z planes in the 1931 ICI system of colorimetry, *J. Opt. Soc. Am.,* 32(3), 168–173, 1942.
5. N. Ahmed and K. R. Rao, *Orthogonal Transforms for Digital Signal Processing,* Springer-Verlag, New York, 1975.
6. M. Anderson, R. Motta, S. Chandrasekar, and M. Stokes, Proposal for a standard default color space for the internet — sRGB, *Proc. IS&T/SID Fourth Color Imaging Conference: Color Science, Systems and Applications,* Scottsdale, AZ, November 1996, 238–246. See also http://www.srgb.com/.
7. W. G. K. Backaus, R. Kliegl, and J. S. Werner, Eds., *Color Vision: Perspectives from Different Disciplines,* Walter de Gruyter, Berlin, 1998.
8. R. Balasubramanian, A printer model for dot-on-dot halftone screens, in J. Bares, Ed., *Proc. SPIE: Color Hard Copy and Graphic Arts IV,* 2413, 356–364, 1995.
9. R. Balasubramanian, Optimization of the spectral neugebauer model for printer characterization, *J. Electronic Imaging,* 8(2), 156–166, 1999.
10. R. Balasubramanian, Reducing the cost of lookup table based color transformations, *J. Imaging Sci. Technol.,* 44(4), 321–327, 2000.
11. R.Balasubramanian, Colorimetric modeling of binary color printers, *Proc. IEEE Int. Conf. on Image Proc.,* November, II-327–330, 1995.
12. J. Bares, Ed., *Proc. SPIE: Color Hard Copy and Graphic Arts,* 1670, February 1992.
13. J. Bares, Ed., *Proc. SPIE: Color Hard Copy and Graphic Arts II,* 1912, February 1993.
14. J. Bares, Ed., *Proc. SPIE: Color Hard Copy and Graphic Arts IV,* 2413, February 1995.
15. J. Bares, Ed., *Proc. SPIE: Color Imaging: Device-Independent Color, Color Hard Copy, and Graphic Arts*, 2658, January 1996.
16. A. Berger-Schunn, *Practical Color Measurement,* John Wiley & Sons, New York, 1994.
17. R. S. Berns, Spectral modeling of a dye diffusion thermal transfer printer. *J. Electronic Imaging,* 2(4), 359–370, 1993.
18. R. S. Berns, Challenges for colour science in multimedia imaging, in L. W. MacDonald and M. R. Luo, Eds., *Colour Imaging: Vision and Technology.* John Wiley & Sons, New York, 1999.

19. R. S. Berns, *Billmeyer and Saltzman's Principles of Color Technology,* 3rd ed., Wiley-Interscience, New York, 2000.
20. R. S. Berns, Visible-spectrum imaging techniques: an overview, in R. Chung and A. Rodrigues, Eds., *Proc. SPIE: The 9th Congress of the International Colour Association,* 4421, June 2001, 475–480.
21. R. S. Berns, M. E. Gorzynski, and R. J. Motta, CRT colorimetry, Part II: Metrology, *Color Res. Appl.,* 18(5), 315–325, 1993.
22. R. S. Berns and F. Imai, Pigment identification of artist materials via multispectral imaging, in *Proc. IS&T/SID Ninth Color Imaging Conference: Color Science, Systems and Applications,* Scottsdale, AZ, November 2001, 209–213.
23. R. S. Berns, R. J. Motta, and M. E. Gorzynski, CRT Colorimetry, Part I: Theory and practice, *Color Res. Appl.,* 18(5), 299–314, 1993.
24. R. S. Berns and K. H. Petersen, Empirical modeling of systematic spectrophotometric errors, *Color Res. Appl.,* 13(4), 243–256, 1988.
25. F. Birren, Ed., *Munsell: A Grammar of Color, A Basic Treatise on the Color System of Albert H Munsell,* Van Nostrand Reinhold, New York, 1969.
26. H. Boll, A color to colorant transformation for a seven ink process, in E. Walowit, Ed., *Proc. SPIE: Device-Independent Color Imaging and Imaging Systems Integration,* 2170, 108–118, 1994.
27. E. G. Boring, The stimulus-error, *Am. J. of Psychol.,* 32, 449–471, 1921.
28. R. W. Boyd, *Radiometry and the Detection of Optical Radiation,* John Wiley & Sons, New York, 1983.
29. D. H. Brainard, Calibration of a computer controlled color monitor, *Color Res. Appl.,* 14(1), 23–34, 1989.
30. D. H. Brainard, Bayesian method for reconstructing color images from trichromatic samples, in *Proc. IS&T's 47th Annual Conference, ICPS'94: The Physics and Chemistry of Imaging Systems,* 2, May 1994, 375–380.
31. D. H. Brainard, Hyper-spectral image data, http://color.psych.upenn.edu/hyperspectral/, 2002.
32. D. H. Brainard and D. Sherman, Reconstructing images from trichromatic samples: from basic research to practical applications, in *Proc. Third IS&T/SID Color Imaging Conference: Color Science, Systems and Applications,* November 1995, 4–10.
33. G. J. Braun, M. D. Fairchild, and F. Ebner, Color gamut mapping in a hue-linearized CIELAB color space, in *Proc. IS&T/SID Sixth Color Imaging Conference: Color Science, Systems and Applications,* Scottsdale, AZ, November 1998, 163–168.
34. W. R. J. Brown and D. L. MacAdam, Visual sensitivities to combined chromaticity and luminance differences, *J. Opt. Soc. Am.,* 39(10), 808–834, 1949.
35. W. Budde, *Optical Radiation Measurements: Physical Detectors of Optical Radiation,* 4th ed., Academic Press, New York, 1983.
36. J. B. Campbell, *Introduction to Remote Sensing,* Guilford Press, New York, 1987.
37. J. A. Castellano, *Handbook of Display Technology,* Academic Press, New York, 1992.
38. J. A. Castellano, Trends in the global CRT market, in *SID Int. Symp. Digest of Tech. Papers,* May 1999, 356–359.
39. CCIR, Encoding parameters of digital television for studios, CCIR Recommendation 601-2, International Radio Consultative Committee (ITU), Geneva, Switzerland, 1990.

40. J. Z. Chang, C. A. Bouman, and J. P. Allebach, Recent results in color calibration using sequential linear interpolation, in *IS&T's 47th Annual Conference, ICPS'94: The Physics and Chemistry of Imaging Systems*, 2, May 1994, 500–505.
41. P. R. Chang and T. H. Hsieh, Constrained nonlinear optimization approaches to color signal separation, *IEEE Trans. Image Proc.*, 4(1), 81–94, 1995.
42. P. Chen and H. J. Trussell, Color filter design for multiple illuminants and detectors, *Proc. Third IS&T/SID Color Imaging Conference: Color Science, Systems and Applications,* November 1995, 67–70.
43. C. D. Child, Discharge from hot CaO, *Phys. Rev.*, 32(5), 492–511, 1911.
44. R. Chung and A. Rodrigues, Eds., *Proc. SPIE: The 9th Congress of the International Colour Association,* 4421, June 2001.
45. *Proc. IS&T/SID Ninth Color Imaging Conference: Color Science, Systems and Applications,* Scottsdale, AZ, November 2001.
46. *Proc. IS&T/SID Fourth Color Imaging Conference: Color Science, Systems and Applications,* Scottsdale, AZ, November 1996.
47. CIE, *Colorimetry,* CIE Publication No. 15.2, Central Bureau of the CIE, Vienna, 1986, 6. The commonly used data on color matching functions is available at the CIE web site at http://www.cie.co.at/.
48. CIE, *Special Metamerism Index: Change in Observer,* CIE Publication No. 80–1989, Central Bureau of the CIE, Vienna, 1989.
49. CIE, *Industrial Color Difference Evaluation,* CIE Publication No. 116–1995, Central Bureau of the CIE, Vienna, 1995.
50. CIE, *The CIE 1997 Interim Colour Appearance Model (Simple Version),* CIECAM97s, Report of the CIE Technical Committee TC1-34, August 1997.
51. F. R. Clapper and J. A. C. Yule, Reproduction of color with halftone images, in *Proc. Seventh Annual Tech. Mtg. TAGA,* May 1955, 1–14.
52. F. J. J. Clarke, R. McDonald, and B. Rigg, Modification to the JPC79 colour-difference formula, *J. Soc. Dyers Colourists,* 100, 128–132, 1984.
53. D. Coco, Breathing life into 3-D humans, *Computer Graphics World,* 18(8), 28–38, 1995.
54. J. Cohen, Dependency of the spectral reflectance curves of the Munsell color chips, *Psychronomic Science,* 1, 369–370, 1964.
55. J. B. Cohen, Color and color mixture: scalar and vector fundamentals, *Color Res. Appl.*, 13(1), 5–39, 1988.
56. J. B. Cohen, Ed., *Visual Color and Color Mixture: The Fundamental Color Space,* University of Illinois Press, Urbana, IL, 2001.
57. J. B. Cohen and W. E. Kappauf, Metameric color stimuli, fundamental metamers, and Wyszecki's metameric blacks, *Am. J. Psychol.*, 95(4), Winter 1982, 537–564.
58. J. B. Cohen and W. E. Kappauf, Color mixture and fundamental metamers: theory, algebra, geometry, application, *Am. J. Psychol.*, 98(2), 171–259, 1985.
59. D. R. Cok, Reconstruction of CCD images using template matching, in *IS&T's 47th Annual Conference, ICPS'94: The Physics and Chemistry of Imaging Systems,* 2, May 1994, 380–385.
60. C. Connolly and T. Fliess, A study of efficiency and accuracy in the transformation from RGB to CIELAB color space, *IEEE Trans. Image Proc.*, 6(7), 1046–1048, 1997.
61. International Color Consortium, International Color Consortium Profile Format, ver. 3.2, November 20, 1995, available by anonymous ftp from ftp.fogra.org.

62. International Color Consortium, Specification ICC.1:2001-04 File Format for Color Profiles, 2001, available online at http://www.color.org/.
63. J. N. Cook, P. A. Sample, and R. N. Weinreb, Solution to spatial inhomogeneity on video monitors, *Color Res. Appl.,* 18(5), 334–340, 1993.
64. W. B. Cowan, An inexpensive scheme for calibration of a color monitor in terms of standard CIE coordinates, *Comp. Graphics,* 17, 315–321, 1983.
65. W. B. Cowan and N. Rowell, On the gun independency and phosphor constancy of colour video monitors, *Color Res. Appl.,* 11 (suppl.) S34–S38, 1986.
66. S. Daly, The visual differences predictor: An algorithm for the assessment of image fidelity, in A. B. Watson, Ed., *Digital Images and Human Vision,* MIT Press, Cambridge, MA, 179–206, 1993.
67. S. Daly, X. Feng, and J. Speigle, A few practical applications that require some of the advanced features of current visual models, in B. E. Rogowitz and T. N. Pappas, Eds., *Proc. SPIE: Human Vision and Electronic Imaging VII,* 4662, January 2002, 70–83.
68. L. E. DeMarsh, T.V. display phosphors/primaries — some history, *SMPTE J.,* 102, 1095–1098, 1993.
69. E. Demichel, in *Procédé,* 26, 17–21 and 26–27, 1924.
70. E. L. Dereniak and D. G. Crowe, *Optical Radiation Detectors,* Wiley, New York, 1984.
71. J. A. Díaz, A. Chiron, and F. Viénot, Tracing a metameric match to individual variations of color vision, *Color Res. Appl.,* 23(6), 379–389, 1998.
72. R. C. Durbeck and S. Sherr, Eds., *Output Hardcopy Devices,* Academic Press, New York, 1988.
73. M. D'Zmura, T. Shen, W. Wu, H. Chen, and M. Viassilou, Contrast gain control for color image quality, in B. E. Rogowitz and T. N. Pappas, Eds., *Proc. SPIE: Human Vision and Electronic Imaging III,* 3299, January 1998, 194–201.
74. P. Emmel, Modèles de Prédiction Couleur Appliqués à L'Impression Jet D'Encre, Ph. D. dissertation, École Polytechnique Fédérale de Lausanne, Switzerland, 1998.
75. P. Emmel and R. D. Hersch, Predicting the spectral behavior of colour printers for transparent inks on transparent support. *IEEE Trans. Pattern Anal. Mach. Intel.,* 19(2), 54–60, 1999.
76. K. Engelhardt and P. Seitz, Optimum color filters for CCD digital cameras, *Optical Eng.,* 32(16), 3015–3023, 1993.
77. R. Eschbach and G. G. Marcu, Eds., *Proc. SPIE: Color Imaging: Device Independent Color, Color Hard Copy, and Applications VII,* 4663, January 2002.
78. Estévez, A better colorimetric standard observer for colorvision studies: the Stiles and Burch 2° color-matching functions, *Color Res. Appl.,* 7, 131, 1982.
79. M. D. Fairchild, Formulation and testing of an incomplete-chromatic-adaptation model, *Color Res. Appl.,* 16(4), 243–250, 1991.
80. M. D. Fairchild, Considering the surround in device independent color imaging, *Color Res. Appl.,* 20(6), 352–363, 1995.
81. M. D. Fairchild, *Color Appearance Models,* Addison-Wesley, Reading, MA, 1998.
82. M. D. Fairchild, A revision of CIECAM97s for practical applications, *Color Res. Appl.,* 26(6), 418–427, 2001.
83. M. D. Fairchild and L. Reniff, Propagation of random errors in spectrophotometric colorimetry, *Color Res. Appl.,* 16(6), 360–367, 1991.

84. M. D. Fairchild and D. Wyble, Colorimetric characterization of the Apple studio display (flat panel LCD), Munsell Color Science Laboratory Technical Report, July 1998.
85. H. S. Fairman, M. H. Brill, and H. Hemmendinger, How the CIE 1931 color-matching functions were derived from Wright-Guild data, *Color Res. Appl.,* 22(1), 11–23, 1997.
86. J. E. Farrell and B. A. Wandell, Scanner linearity, *J. Electronic Imaging,* 2(3), 225–230, 1993.
87. O. D. Faugeras, Digital color image processing within the framework of a human visual model, *IEEE Trans. Acoust. Speech Sign. Proc.,* 27(4), 380–393, 1979.
88. G. D. Finlayson, M. S. Drew, and B. V. Funt, Color constancy: generalized diagonal transforms suffice, *J. Opt. Soc. Am. Abstr.,* 11(11), 3011–3019, 1994.
89. G. D. Finlayson, M. S. Drew, and B. V. Funt, Spectral sharpening: Sensor transformations for improved color constancy, *J. Opt. Soc. Am. Abstr.,* 11(5), 1553–1563, 1994.
90. G. D. Finlayson and B. V. Funt, Coefficient channels: Derivation and relationship to other theoretical studies, *Color Res. Appl.,* 21(2), 87–96, 1996.
91. T. G. Fiske and L. D. Silverstein, Characterizations of viewing-angle-dependent colorimetric and photometric performance of color LCDs, in *SID Int. Symp. Digest of Tech. Papers,* 565–568, 1993.
92. J. D. Foley, A. van Dam, and J. F. Hughes, *Computer Graphics: Principles and Practice,* 2nd ed., Addison-Wesley, Reading, MA, 1993.
93. D. A. Forsyth, A novel algorithm for color constancy, *Int. J. Computer Vision,* 5(1), 5–36, 1990.
94. Foveon, Foveon X3 technology, online at http://www.foveon.com.
95. W. Frei and B. Baxter, Rate-distortion coding simulation for color images, *IEEE Trans. Comm.,* COM-25(11), 1977.
96. A. Friedmam, *The Foundations of Modern Analysis,* Dover Publications, New York, 1982.
97. B. V. Funt, F. Ciurea, and J. J. McCann, Tuning retinex parameters, in B. E. Rogowitz and T. N. Pappas, Eds., *Proc. SPIE: Human Vision and Electronic Imaging VII,* 4662, January 2002.
98. R. S. Gentile, J. P. Allebach, and E. Walowi, A comparison of techniques for color gamut mismatch compensation, in B. E. Rogowitz, Ed., *Proc. SPIE: Human Vision, Visual Processing, and Digital Display,* 1077, 342–354, 1989.
99. R. S. Gentile, J. P. Allebach, and E. Walowit, Quantization of color images based on uniform color spaces, *J. Imaging Technol.,* 16, 11–21, 1990.
100. G. A. Gescheider, *Psychophysics: The Fundamentals,* 3rd ed., Lawrence Erlbaum Assoc., Mahwah, NJ, 1997.
101. G. H. Golub and C. F. Van Loan, *Matrix Computations,* 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.
102. J. J. Gordon and R. A. Holub, On the use of linear transformations for scanner calibration, *Color Res. Appl.,* 18(3), 218–219, 1993.
103. E. Granger, Uniform color space as a function of spatial frequency, in *Proc. IS&T Seventh Int. Conf. on Advances in Non-Impact Printing Technology,* 1991, 309–322.
104. E. M. Granger, Press controls for extra-trinary printing, in J. Bares, Ed., *Proc. SPIE: Color Imaging: Device-Independent Color, Color Hard Copy, and Graphic Arts*, 2658, January 1996, 147–150.

105. H. G. Grassmann, Zur theorie der farbenmischung, Poggendorf, *Annalen der Physik und Chemie*, 89, 69–84, 1853, translation titled Theory of compound colors, in *Philosophic Magazine,* 4(7), 254–264, 1854, reprinted in References 188 and 190.
106. R. O. Green, C. M. Sarture, C. J. Chovit, J. A. Faust, P. Hajek, and H. I. Novak, AVIRIS: a new approach to earth remote sensing, *Optics Photonics News,* 6(1), 30–33, 1995.
107. S. Gregory, R. Poe, and D. Walker, Communicating color appearance with the ICC color profile, in *Proc. IS&T/SID Second Color Imaging Conference: Color Science, Systems and Applications,* 170–174, Scottsdale, AZ, November 15–18, 1994.
108. F. Grum and C. J. Bartleson, Eds., *Optical Radiation Measurements: Color Measurement,* Vol. 2, Academic Press, New York, 1983.
109. F. Grum and R. J. Becherer, *Optical Radiation Measurements: Radiometry,* Vol. 1, Academic Press, New York, 1979.
110. J. Guild, The colorimetric properties of the spectrum, *Philos. Trans. Royal Soc. London A*, 230, 149–178, 1931.
111. S. Gustavson, Color gamut of halftone reproduction, *J. Imaging Sci. Technol.,* 41(2), 283–290, 1997.
112. H. Haneishi, T. Hirao, A. Shimazu, and Y. Mikaye, Colorimetric precision in scanner calibration using matrices, in *Proc. Third IS&T/SID Color Imaging Conference: Color Science, Systems and Applications,* November 1995, 106–108.
113. R. A. Hann, Thermal dye diffusion printing, in J. Bares, Ed., *Proc. SPIE: Color Hard Copy and Graphic Arts II,* 1912, February 1993, 252–260.
114. R. A. Hann and N. C. Beck, Dye diffusion thermal transfer (D2C2) color printing, *J. Imaging Technol.,* 16(6), 238–241, 1990.
115. J. Y. Hardebebrg, F. Schmitt, and H. Brettel, Multispectral image capture using a tunable filter, in R. Eschbach and G. G. Marcu, Eds., *Proc. SPIE: Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts V,* 3963, January 2000, 23–28.
116. J. Y. Hardeberg, F. Schmitt, H. Brettel, J. P. Crettez, and H. Maitre, Multispectral image acquisition and simulation of illuminant changes, in L. W. MacDonald and M. R. Luo, Eds., *Colour Imaging: Vision and Technology,* John Wiley & Sons, New York, 1999.
117. J. E. Hardis, Improving color measurements of displays, in J. Bares, Ed., *Proc. SPIE: Color Imaging: Device-Independent Color, Color Hard Copy, and Graphic Arts*, 2658, January 1996, 182–191.
118. Y. Hashimoto, M. Yamamoto, and T. Asaid, Cameras and display systems, *Proc. IEEE*, 83(7), 1032–1043, 1995.
119. C. Heinmüller, G. Haas, and P. M. Knoll, Design of in-plane-compensation foils for viewing-angle enhancement, in *SID Int. Symp. Digest of Tech. Papers,* May 1999, 90–93.
120. E. Hering, *Outlines of a Theory of the Light Sense,* Harvard University Press, Cambridge, MA, 1964, translated from German by L. M. Hurvich and D. Jameson.
121. B. Hill, Optimization of total multispectral imaging systems: best spectral match versus least observer metamerism, in R. Chung and A. Rodrigues, Eds., *Proc. SPIE: The 9th Congress of the International Colour Association,* 4421, June 2001, 481–486.

122. J. Ho, B. V. Funt, and M. S. Drew, Separating a color signal into illumination and surface reflectance components: Theory and applications, *IEEE Trans. Pattern Anal. Mach. Intel.,* 12(10), 966–977, 1990.
123. A. Høard, L. Sivik, and G. Tonnquist, NCS natural color system — from concept to research and applications I, *Color Res. Appl.,* 21(3), 180–205, 1996.
124. A. Høard, L. Sivik, and G. Tonnquis, NCS natural color system — from concept to research and applications II, *Color Res. Appl.,* 21(3), 206–220, 1996.
125. B. K. P. Horn, Exact reproduction of color images, *Comp. Vis., Graphics and Image Proc.,* 26, 135–167, 1984.
126. J. J. Y. Huang and P. M. Schultieiss, Block quantization of correlated Gaussian random variables, *IEEE Trans. Comm. Sys.,* CS-11, 289–296, 1963.
127. P. C. Hung, Colorimetric calibration for scanners and media, *Proc. SPIE,* 1448, 164–174, 1991.
128. P. C. Hung, Colorimetric calibration in electronic imaging devices using a look-up table model and interpolations, *J. Electronic Imaging,* 2(1), January 53–61, 1993.
129. P. C. Hung and R. S. Berns, Determination of constant hue loci for a CRT gamut and their predictions using color appearance spaces, *Color Res. Appl.,* 20(5), 285–295. 1995.
130. R. W. G. Hunt, *The Reproduction of Colour,* 3rd. ed., John Wiley & Sons, New York, 1975.
131. R. W. G. Hunt, Sky-blue pink, *Color Res. Appl.,* 1(1), 11–16, 1976.
132. R. W. G. Hunt, The specification of colour appearance, I, Concepts and terms, *Color Res. Appl.,* 2(2), 55–68, 1977.
133. R. W. G. Hunt, Colour terminology, *Color Res. Appl.,* 3(2), 79–87, 1978.
134. R. W. G. Hunt, *The Reproduction of Colour in Photography, Printing, and Television,* 4th ed., Fountain Press, Tolworth, England, 1987.
135. R. W. G. Hunt, *Measuring Colour,* 2nd ed., Ellis Horwood, New York, 1991.
136. R. W. G. Hunt, *The Reproduction of Colour,* 5th ed., Fountain Press, Tolworth, England, 2000.
137. R. M. Adams II and J. B. Weisberg, Eds., *The GATF Practical Guide to Color Management,* GATF Press, Pittsburgh, PA, 1998.
138. F. Imai and R. S. Berns, Optimization of total multispectral imaging systems: best spectral match versus least observer metamerism, in R. Chung and A. Rodrigues, Eds., *Proc. SPIE: The 9th Congress of the International Colour Association,* 4421, June 2001, 504–507.
139. M. Inui, Fast algorithms for computing color gamuts, *Color Res. Appl.,* 18(5), 341–348, 1993.
140. ITU, Basic parameter values for the HDTV standard for the studio and for international programme exchange, ITU-R Recommendation BT 709, ITU, Geneva, Switzerland, April 1990, formerly CCIR Rec. 709.
141. H. E. Ives, The transformation of color-mixture equations from one system to another, *J. Franklin Inst.,* 16, 673–701, 1915.
142. A. K. Jain, *Fundamentals of Digital Image Processing,* Prentice-Hall, Englewood Cliffs, NJ, 1989.
143. T. H. James, Ed., *The Theory of Photographic Process,* 4th ed., MacMillan, New York, 1977.
144. T. E. Jenkins, *Optical Sensing Techniques and Signal Processing,* Prentice-Hall, NJ, 1987.

145. D. J. Jobson, Z. Rahman, and G. A. Woodell, A multi-scale retinex for bridging the gap between color images and the human observation of scenes, *IEEE Trans. Image Proc.,* 6(7), 965–976, 1997.
146. D. J. Jobson, Z. Rahman, and G. A. Woodell, Properties and performance of a center/surround retinex, *IEEE Trans. Image Proc.,* 6(3), 451–462, 1997.
147. T. Johansson, Färg. Lindfors Bokförlag, AB, Sweden, in Swedish, 1937.
148. J. L. Johnson, *Principles of Nonimpact Printing,* 2nd ed., Palatino Press, Irvine, CA, 1992.
149. I. T. Jolliffe, *Principal Components Analysis,* Springer-Verlag, Berlin, 1986.
150. D. B. Judd, Reduction of data on mixture of color stimuli, *Bureau of Standards J. Research,* 4, 515–548, 1930.
151. D. B. Judd, Chromatic sensibility to stimulus differences, *J. Opt. Soc. Am.,* 22(2), 72–108, 1932.
152. D. B. Judd, A maxwell triangle yielding uniform chromaticity scales, *J. Opt. Soc. Am.,* 25(1), 24–35, 1935.
153. P. K. Kaiser and R. M. Boynton, *Human Color Vision,* 2nd ed., Optical Society of America, Washington, D.C., 1996.
154. D. Kalra, GamOpt: a tool for visualization and optimization of gamuts, in J. Bares, ed., *Proc. SPIE: Color Hard Copy and Graphic Arts III,* 2171, February 1994, 299–309.
155. E. Kaneko, *Liquid Crystal TV Displays: Principles and Applications of Liquid Crystal Displays,* KTK Scientific Publishers, Tokyo, 1987.
156. S. Kaneko, Y. Hirai, and K. Sumiyoshi, Wide-viewing angle improvements for AMLCDs, in *SID Int. Symp. Digest of Tech. Papers,* May 1993, 265–268.
157. H. Kang, Printer-related color processing techniques, in J. Bares, Ed., *Proc. SPIE: Color Hard Copy and Graphic Arts IV,* 2413, February 1995, 410–419.
158. H. R. Kang, Color scanner calibration, *J. Imaging Sci. and Technol.,* 36(2), 162–170, 1992.
159. H. R. Kang, *Color Technology for Electronic Imaging Devices,* SPIE, Bellingham, WA, 1997.
160. H. R. Kang and P. G. Anderson, Neural network applications to the color scanner and printer calibrations, *J. Electronic Imaging,* 1(2), 125–134, 1992.
161. J. Chen and K. C. Kang, J. DelPico, H. Seiberle, and M. Schad, Wide-viewing-angle photoaligned plastic films for TN-LCDs, in *SID Int. Symp. Digest of Tech. Papers,* 98–101, 1999.
162. M. A. Karim, Ed., *Electro-Optical Displays,* Marcel Dekker, New York, 1992.
163. J. M. Kasson and W. Plouffe, An analysis of selected computer interchange color spaces, *ACM Trans. Graphics,* 11(4), 373–405, 1992.
164. D. H. Kelly, Spatial and temporal interactions in color vision, *J. Imaging Technol.,* 15(2), 82–89, 1989.
165. F. König and W. Praefcke, A multispectral scanner, in L. W. MacDonald and M. R. Luo, Eds., *Colour Imaging: Vision and Technology,* John Wiley & Sons, New York, 1999.
166. F. Kretz, Subjectively optimal quantization of pictures, *IEEE Trans. Comm.,* COM-23(11), 1288–1292, 1975.
167. J. A. von Kries, Die Gesichtsempfindungen, in W. Nagel, ed., *Handbuch der Physiologie der Menschen, Tiel III,* 3, Vieweg, Brunswick, 1905, 109–282, reprinted under the English title, Influence of adaptation on the effects produced by luminous stimuli, in D.L. MacAdam, Ed., *Sources of Color Science,* 1970, MIT Press, Cambridge, MA, 1970, 121–126.

168. P. Kubelka and F. Munk, Ein Beitrag zur Optik der Farbanstriche, *Z. Tech. Phys.,* 11a, 593–601, 1931.
169. R. G. Kuehni, Towards an improved uniform color space, *Color Res. Appl.,* 24(4), 253–265, 1999.
170. R. G. Kuehni, CIEDE2000, milestone of final answer, *Color Res. Appl.,* 27(2), 126–127, 2002.
171. M. Kulkarni and M. Grant, Color management and film recorders, *Photographic Processing,* January, 52–53, 1995.
172. Y. Kwak and L. W. MacDonald, Accurate prediction of colours on liquid crystal displays, in *Proc. IS&T/SID Ninth Color Imaging Conference: Color Science, Systems and Applications,* Scottsdale, AZ, November 2001, 355–359.
173. E. H. Land, The retinex theory of color vision, *Sci. Am.,* 237(6), 108–129, 1977.
174. E. H. Land and J. J. McCann, Lightness and retinex theory, *J. Opt. Soc. Am.,* 61(1), 1–11, 1971.
175. I. Langmuir, The effect of space charge and residual gases on thermionic current in high vacuum, *Phys. Rev., Second Series,* 2(6), 450–486, 1913.
176. C. J. Li, M. R. Luo, and R. W. G. Hunt, A revision of the CIECAM97s model, *Color Res. Appl.,* 26(6), 260–266, 2000.
177. J. O. Limb, C. B. Rubinstein, and J. E. Thompson, Digital coding of color video signals — a review, *IEEE Trans. Comm.,* COM-25(11), 1349–1385, 1977.
178. J. Lubin, A visual discrimination model for imaging system design and evaluation, in E. Peli, Ed., *Vision Models for Target Detection and Recognition,* World Scientific, River Edge, NJ, 245–283, 1995.
179. M. R. Luo, G. Cui, and B. Rigg, The development of the CIE 2000 colour-difference formula: CIEDE2000, *Color Res. Appl.,* 26(5), 340–350, 2001.
180. M. R. Luo, G. Cui, and B. Rigg, Further comments on CIEDE2000, *Color Res. Appl.,* 27(2), 127–128, 2002.
181. M. R. Luo and R. W. G. Hunt, The structure of the CIE 1997 colour appearance model (CIECAM97s), *Color Res. Appl.,* 23(3),138–146, 1998.
182. M. R. Luo and B. Rigg, BFD(l:c) colour-difference formula, Part I — development of the formula, *J. Soc. Dyers Colourists,* 103, 86–94, 1987.
183. M. R. Luo and B. Rigg, BFD(l:c) colour-difference formula, Part II — performance of the formula, *J. Soc. Dyers Colourists,* 103, 126–132, 1987.
184. R. Luther, Aus Dem Gebiet der Farbreizmetrik, *Z. Tech. Phys.,* 8, 540–558, 1927.
185. D. L. MacAdam, Visual sensitivities to color differences in daylight, *J. Opt. Soc. Am.,* 32(5), 247–274, 1942.
186. D. L. MacAdam, Uniform color scales, *J. Opt. Soc. Am.,* 64, 1619–1702, 1974.
187. D. L. MacAdam, Colorimetric data for samples of OSA uniform color scales, *J. Opt. Soc. Am.,* 68, 121–130, 1978.
188. D. L. MacAdam, Ed., *Sources of Color Science,* MIT Press, Cambridge, MA, 1970.
189. D. L. MacAdam, *Color Measurement: Theme and Variations,* 2nd ed., Springer-Verlag, New York, 43–45, 1981.
190. D. L. MacAdam, Ed., *Selected Papers on Colorimetry — Fundamentals*, SPIE Optical Engineering Press, Bellingham, WA, 1993.
191. L. W. MacDonald and M. R. Luo, Eds., *Colour Imaging: Vision and Technology,* John Wiley & Sons, Ltd., New York, 1999.
192. M. Mahy, L. Van Eyckden, and A. Oosterlinck, Evaluation of uniform color spaces developed after the adoption of CIELAB and CIELUV, *Color Res. Appl.,* 19(2), 105–121, 1994.

193. H. Maître, F. Schmitt, J. P. Crettez, Y. Wu, and J. Y. Hardeberg, Spectrophotometric image analysis of fine art paintings, in *Proc. IS&T/SID Fourth Color Imaging Conference: Color Science, Systems and Applications,* Scottsdale, AZ, November 1996, 50–53.
194. L. T. Maloney, Evaluation of linear models of surface reflectance with a small number of parameters, *J. Opt. Soc. Am. Abstr.,* 3, 1673–1683, 1986.
195. L. T. Maloney and B. A. Wandell, Color constancy: A method for recovering surface reflectance, *J. Opt. Soc. Am. Abstr.,* 3(1), 29–33, 1986.
196. V. Mani, Calibration of color monitors, M.S. thesis, North Carolina State University, Raleigh, NC, 1991.
197. G. Marcu, W. Chen, P. Graffagnino, and O. Andrade, Color characterization issues for TFTLCD displays, in R. Eschbach and G. G. Marcu, Eds., *Proc. SPIE: Color Imaging: Device Independent Color, Color Hard Copy, and Applications VII,* 4663, January 2002, 187–198.
198. D. H. Marimont and B. A. Wandell, Linear models of surface and illuminant spectra, *J. Opt. Soc. Am. Abstr.,* 9(11), 1905–1913, 1992.
199. S. Matsumoto, Ed., *Electronic Display Devices,* John Wiley & Sons, New York, 1990.
200. James Clerk Maxwell, The diagram of colors, *Transactions of the Royal Society of Edinburgh,* 21, 1857, 275–298, reprinted in References 188 and 190.
201. J. C. Maxwell, Theory of compound colors and the relations to the colors of the spectrum, *Proc. Royal Society of London,* 10, 404–409, 1860, reprinted in References 188 and 190.
202. C. S. McCamy, Correlated color temperature as an explicit function of chromaticity coordinates, *Color Res. Appl.,* 17(2), 142–144, 1992.
203. J. J. McCann, Capturing a black cat in shade: the past and present of retinex color appearance models, in B. E. Rogowitz and T. N. Pappas, Eds., *Proc. SPIE: Human Vision and Electronic Imaging VII,* 4662, 331–340, 2002.
204. R. McCluney, *Introduction to Radiometry and Photometry,* Artech House, Boston, MA, 1994.
205. R. McDonald, Acceptability and perceptibility decisions using the CMC color difference formula, *TCC,* 20(6), 31–37, 1988. See also errata for important corrections to this paper.
206. J. Meyer and B. Barth, Color gamut mapping for hard copy, in *SID Digest,* 86–89, 1989.
207. J. L. Mitchell, C. Fogg, D. J. LeGall, and W. B. Pennebaker, Eds., *MPEG Digital Video Compression Standard,* Chapman & Hall, New York, 1997.
208. A. H. Munsel, *A Color Notation,* 1st ed., Munsell Color Co., Baltimore, MD, 1905.
209. *Munsell Book of Color,* Munsell Color Co., Baltimore, MD, 1929–present.
210. K. Nassau, *The Physics and Chemistry of Color: The Fifteen Causes of Color,* John Wiley & Sons, New York, 1983.
211. Y. Nayatani, K. Takahama, and H. Sobagaki, Physiological causes of variations of color-matching functions, *Color Res. Appl.,* 13, 289–297, 1988.
212. A. Netravali and B. Haskell, Eds., *Digital Pictures, Representation, Compression, and Standards*, Plenum Press, New York, 1995.
213. H. E. J. Neugebauer, Die theoretischen Grundlagen des Mehrfarbenbuchdrucks, *Zeitschrift für wissenschaftliche Photographie Photophysik und Photochemie,* 36(4), 73–89, 1937, reprinted in K. Sayangi, Ed., *Proc. SPIE: Neugebauer Memorial Seminar on Color Reproduction,* 1184, SPIE, Bellingham, WA, 1989.

214. H. E. J. Neugebauer, Quality factor for filters whose spectral transmittances are different from color mixture curves, and its application to color photography, *J. Opt. Soc. Am.*, 46(10), 821–824, 1956.
215. I. Newton, New theory of light and colors, *Philos. Trans. Roy. Soc. London*, 80, 3075–3087, 1671/72, reprinted in References 188 and 190.
216. I. Newton, *Opticks, or, a treatise of the reflections, refractions, inflections & colours of light*, 4th ed., Dover Publications, New York, 1952, based on the 4th ed., London, 1730.
217. M. Nier and M. E. Courtot, Eds., *Standards for electronic imaging systems*, proceedings of a conference held 28 February 1–March 1991, San Jose, CA, CR37 of *Critical Reviews of Optical Science and Technology*, SPIE, Bellingham, WA, 1991.
218. S. I. Nin, J. M. Kasson, and W. Plouffe, Printing CIELAB images on a CMYK printer using tri-linear interpolation, in J. Bares, Ed., *Proc. SPIE: Color Hard Copy and Graphic Arts*, 1670, February 1992, 356–364.
219. A. D. North and M. D. Fairchild, Measuring color-matching functions, Part I, *Color Res. Appl.*, 18(3), 155–162, 1993.
220. A. D. North and M. D. Fairchild, Measuring color-matching functions, Part II, New data for assessing observer metamerism, *Color Res. Appl.*, 20(1), 29–35, 1995.
221. NTSC, NTSC signal specifications, *Proc. IRE*, IRE-42(1), 17–19, 1954.
222. B. M. Oliver, Tone rendition in television, *Proc. IRE*, 38, 1288–1300, 1950.
223. V. Ostromoukhov, Chromaticity gamut enhancement by heptatone multicolor printing, in R. J. Motta and H. A. Berberian, Eds., *Proc. SPIE: Device-Independent Color Imaging and Imaging Systems Integration*, 1909, February 1993, 139–150.
224. Pantone, Inc., Pantone Matching System, information available at http://www.pantone.com.
225. N. W. Parker, An analysis of the necessary receiver decoder corrections for color receiver operation with non-standard primaries, *IEEE Trans. Broadcast and TV Rec.*, BTR-12(1), 23–32, 1966.
226. K. A. Parulski, L. J. D'Luna, B. L. Benamati, and P. R. Shelley, High-performance digital color video camera, *J. Electronic Imaging*, 1(1), 35–45, 1992.
227. R. Patterson, Gamma correction and tone reproduction in scanned photographic images, *SMPTE J.*, 103, 377–385, 1994.
228. W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*, Van Nostrand Reinhold, New York, 1993.
229. M. R. Pointer, A comparison of the CIE 1976 colour spaces, *Color Res. Appl.*, 6(2), 108–118, 1981.
230. J. Pollack, Sharp microelectronics' approach to new-generation AMLCDs, *Information Display*, 15(2), 16–20, 1999.
231. C. Poynton, Frequently asked questions about color, available at http://www.poynton.com/PDFs/ColorFAQ.pdf.
232. C. A. Poynton, Gamma and its disguises: the nonlinear mappings of intensity in perception, CRTs, film, and video, *SMPTE J.*, 102, 1099–1108, 1993.
233. W. K. Pratt, Spatial transform coding of color images, *IEEE Trans. Commun. Tech.*, COM-19(12), 980–992, 1971.
234. W. K. Pratt, *Digital Image Processing*, 2nd ed., John Wiley & Sons, New York, 1991.

235. S. Quan, Evaluation and Optimal Design of Spectral Sensitivities for Digital Color Imaging, Ph.D. dissertation, Rochester Institute of Technology, April 2002.
236. S. Quan and N. Ohta, Evaluating quality factors of hypothetical spectral sensitivities, in *Final Prog. and Proc. Image Processing, Image Quality, Image Capture, Systems Conference*, Portland, OR, March 2000, IS&T, 37–42.
237. S. Quan, N. Ohta, R. Berns, and N. Katoh, Optimal design of camera spectral sensitivity functions based on practical color filter components, in *Proc. IS&T/SID Ninth Color Imaging Conference: Color Science, Systems and Applications,* Scottsdale, AZ, November 2001, 277–282.
238. S. Quan, N. Ohta, M. Rosen, and N. Katoh, Fabrication tolerance and optimal design of spectral sensitivities for color imaging devices, in *Final Prog. and Proc. Image Processing, Image Quality, Image Capture, Systems Conference*, Montréal, Canada, April 22–25, 2001, IS&T, 277–282.
239. Z. Rahman, D. J. Jobson, and G. A. Woodell, Retinex processing for automatic image enhancement, in B. E. Rogowitz and T. N. Pappas, Eds., *Proc. SPIE: Human Vision and Electronic Imaging VII,* 4662, January 2002, 390–401.
240. S. A. Rajala and A. P. Kakodkar, Interpolation of color data, in *Proc. IS&T/SID Color Imaging Conference: Transforms and Portability of Color*, November 1993, 180–183.
241. I. Rezanka, Thermal ink jet: a review, in J. Bares, Ed., *Proc. SPIE: Color Hard Copy and Graphic Arts,* 1670, February 1992, 192–200.
242. W. Rhodes, Fifty years of the Neugebauer equations, in K. Sayangi, Ed., *Proc. SPIE: Neugebauer Memorial Seminar on Color Reproduction,* 1184, December 1989, 7–18.
243. W. L. Rhodes and C. M. Hains, The influence of halftone orientation on color gamut and registration sensitivity, in *Proc. IS&T's 46th Annual Conference,* May 1993, 180–182.
244. D. Rich, Euclidean color spaces with logarithmic compression: a comment on Knud Thomsen's note, *Color Res. Appl.,* 25(4), 293, 2000.
245. A. R. Robertson, The CIE 1976 color-difference formulae, *Color Res. Appl.,* 2(1), 7–11, 1977.
246. A. R. Robertson, The future of color science, *Color Res. Appl.,* 7(1), 16–18, 1982.
247. B. Robertson, Toy Story: a triumph of animation, *Computer Graphics World,* 18(8), 28–38, 1995.
248. M. Rodriguez, A graphics arts perspective on RGB-to-CMYK conversion, *Proc. IEEE Int. Conference on Image Proc.*, November 1995, II-319–322.
249. M. A. Rodriguez and T. G. Stockham, Producing colorimetric data from densitometric scans, in J. P. Allebach and B. E. Rogowitz, Eds., *Proc. SPIE: Human Vision, Visual Processing, and Digital Display IV,* 1913, February 1993, 413–418.
250. B. E. Rogowitz and T. N. Pappas, Eds., *Proc. SPIE: Human Vision and Electronic Imaging VII,* 4662, January 2002.
251. R. Rolleston, Visualization of colorimetric calibration, in J. Bares, Ed., *Proc. SPIE: Color Hard Copy and Graphic Arts II,* 1912, February 1993, 299–309.
252. R. Rolleston and R. Balasubramanian, Accuracy of various types of neugebauer models, in *Proc. IS&T/SID Color Imaging Conference: Transforms and Portability of Color*, 32–37, 1993.
253. A. Ryer, *Light Measurement Handbook,* 1997, International Light, Newburyport, MA, available at http://www.intl-light.com/.

254. O. Sahni, Color printing technologies, in D. B. Dove, T. Abe, and J. Heinzl, Eds., *Proc. SPIE: Printing Technologies for Images, Gray Scale, and Color,* 1458, February 1991, 4–16.
255. Kazuo Sayangi, Ed., *Proc. SPIE: Neugebauer Memorial Seminar on Color Reproduction,* 1184, December 1989, 14–15.
256. J. Schanda, Current CIE work to achieve physiologically-correct color metrics, in W. G. K. Backaus, R. Kliegl, and J. S. Werner, Eds., *Color Vision: Perspectives from Different Disciplines,* Walter de Gruyter, Berlin, 307–318, 1998.
257. W. F. Schreiber, *Fundamentals of Electronic Imaging Systems: Some Aspects of Image Processing,* 3rd ed., Springer Verlag, Berlin, 1993.
258. E. W. H. Selwyn, A theory of graininess, *Phot. J.,* 73, 571, 1935.
259. E. W. H. Selwyn, Experiments on the nature of graininess, *Phot. J.,* 79, 513, 1939.
260. G. Sharma, Color Scanner Characterization, Performance Evaluation, and Design, Ph.D. dissertation, North Carolina State University, Raleigh, NC, August 1996.
261. G. Sharma, Set theoretic estimation for problems in subtractive color, *Color Res. Appl.,* 25(4), 333–348, 2000.
262. G. Sharma, Target-less scanner color calibration, *J. Imaging Sci. Technol.,* 44(4), 301–307, 2000.
263. G. Sharma, LCDs versus CRTs: color-calibration and gamut considerations, *Proc. IEEE,* 90(4), 605–622, 2002, special issue on flat panel display technologies.
264. G. Sharma and H. J. Trussell, Decomposition of fluorescent illuminant spectra for accurate colorimetry, *Proc. IEEE Int. Conf. Image Proc. 1994,* November 1994, 1002–1006.
265. G. Sharma and H. J. Trussell, Set theoretic estimation in color scanner characterization, *J. Electronic Imaging,* 5(4), 479–489, 1996.
266. G. Sharma and H. J. Trussell, Digital color imaging, *IEEE Trans. Image Proc.,* 6(7), 901–932, 1997.
267. G. Sharma and H. J. Trussell, Figures of merit for color scanners, *IEEE Trans. Image Proc.,* 6(7), 990–1001, 1997.
268. G. Sharma, H. J. Trussell, and M. J. Vrhel, Optimal non-negative color scanning filters, *IEEE Trans. Image Proc.*, 7(1), 129–133, 1998.
269. G. Sharma, M. J. Vrhel, and H. J. Trussell, Color imaging for multimedia, *Proc. IEEE,* 86(6), 1088–1108, 1998.
270. G. Sharma and S. Wang, Spectrum recovery from colorimetric data for color reproductions, in R. Eschbach and G. G. Marcu, Eds., *Proc. SPIE: Color Imaging: Device Independent Color, Color Hard Copy, and Applications VII,* 4663, January 2002, 8–14.
271. G. Sharma, S. Wang, D. Sidavanahalli, and K. T. Knox, The impact of UCR on scanner calibration, *Final Prog. and Proc. IS&T's PICS Conference*, Portland, OR, May 1998, 121–124.
272. S. Sherr, *Electronic Displays,* 2nd ed., 1993, John Wiley & Sons, New York.
273. *SID Int. Symp. Digest of Tech. Papers,* May 1993.
274. *SID Int. Symp. Digest of Tech. Papers,* May 1999.
275. L. Sivik, Systems for descriptive colour notations — implications of definitions and methodology, *Fabre,* 40(1), 37–49, 1994.
276. B. Sluban, Comparison of colorimetric and spectrophotometric algorithms for computer match prediction, *Color Res. Appl.,* 18(2), 74–79, 1993.

277. V. C. Smith and J. Pokorny, Chromatic-discrimination axes, CRT phosphor spectra, and individual variation in color vision, *J. Opt. Soc. Am. Abstr.,* 12(1), 27–35, 1995.
278. V. C. Smith and J. Pokorny, The design and use of a cone chromaticity space: A tutorial, *Color Res. Appl.,* 21(5), 375–383, 1996.
279. SMPTE, Color Temperature for Color Television Studio Monitors, SMPTE Recommended Practice RP 37–1969, Society for Motion and Television Pictures, White Plains, NY, July 1969.
280. SMPTE, Color Monitor Colorimetry, SMPTE Recommended Practice RP 145–1987, Society for Motion and Television Pictures, White Plains, NY, June 1987.
281. R. Sobol, Improving the retinex algorithm for rendering wide dynamic range photographs, in B. E. Rogowitz and T. N. Pappas, Eds., *Proc. SPIE: Human Vision and Electronic Imaging VII,* 4662, January 2002, 341–348.
282. D. L. Spooner, Translucent blurring errors in small area reflectances spectrophotometric and densitometric measurements, in *Proc. Tech. Assoc. Graphic Arts Annual Meeting,* Rochester, NY, May 1991, 130–143.
283. sRGB website, http://www.srgb.com/.
284. W. S. Stiles and J. M. Burch, N.P.L. colour-matching investigation: final report, *Opt. Acta,* 6, 1–26, 1959.
285. A. Stockman, L. T. Sharpe, and C. C. Fach, The spectral sensitivity of the human short-wavelength cones, *Vision Res.,* 39, 2901–2927, 1999.
286. M. C. Stone, W. B. Cowan, and J. C. Beatty, Color gamut mapping and the printing of digital color images, *ACM Trans. Graphics,* 7(4), 249–292, 1988.
287. P. H. Swain and S. M. Davis, Eds., *Remote Sensing: The Quantitative Approach,* McGraw-Hill, New York, 1978.
288. *Specifications Web Offset Publications (SWOP),* 9th ed., 60 East 42nd Street, Suite 721, New York, 2001, http://www.swop.org/.
289. Natural Color System, *Colour Notation System,* Swedish Standards Institute, Stockholm, 2nd ed., 1990, additional information available at http://www.ncscolor.com.
290. E. Taft and J. Walden, Eds., *PostScript Language Reference Manual,* 2nd ed., Addison-Wesley, Reading, MA, 1990.
291. L. E. Tannas, Ed., *Flat Panel Displays and CRTs,* Van Nostrand Reinhold, New York. 1985.
292. L. A. Taplin and R. S. Berns, Spectral color reproduction based on a six-color inkjet output system, in *Proc. IS&T/SID Ninth Color Imaging Conference: Color Science, Systems and Applications,* Scottsdale, AZ, November 2001, 209–213.
293. D. Taubman, Generalized Weiner reconstruction of images from colour sensor data using a scale invariant prior, in *Proc. IEEE Int. Conf. Image Proc., III,* September 2000.
294. W. Thomas, Ed., *SPSE Handbook of Photographic Science and Engineering*, Wiley Interscience, New York, 1973.
295. K. Thomsen, A Euclidean color space in high agreement with the CIE94 color-difference formula, *Color Res. Appl.*, 25(1), 64–65, 2000.
296. E. B. Titchener, *A Textbook of Psychology,* MacMillan, New York, 1905.
297. S. Tominaga, N. Tanaka, and T. Matsumoto, Recording and rendering for art paintings based on multiband data, in R. Eschbach and G. G. Marcu, Eds., *Proc. SPIE: Color Imaging: Device Independent Color, Color Hard Copy, and Applications VII,* 4663, January 2002, 27–34.

298. D. Travis, *Effective Color Displays: Theory and Practice,* Academic Press, San Diego, 1991.
299. H. J. Trussell, Application of set theoretic models to color systems, *Color Res. Appl.,* 16(1), 31–41, 1991.
300. H. J. Trussell, DSP solutions run the gamut for color systems, *IEEE Sig. Proc. Mag.,* 10(2), 8–23, 1993.
301. H. J. Trussell and R. E. Hartwig, Mathematics of demosaicking, *IEEE Trans. Image Proc.,* 11(4), 485–492, 2002.
302. H. J. Trussell and M. Kulkarni, Estimation of color under fluorescent illuminants, in *Proc. IEEE Int. Conf. Image Proc. 1994,* III1006–1010.
303. H. J. Trussell and M. S. Kulkarni, Sampling and processing of color signals, *IEEE Trans. Image Proc.,* 5, 677–681, 1996.
304. H. J. Trussell and J. R. Sullivan, A vector-space approach to color imaging systems, in K. S. Pennington, Ed., *Proc. SPIE: Image Processing Algorithms and Techniques,* 1244, 264–271, 1990.
305. D. Y. Tzeng and R. S. Berns, Spectral-based six-color separation minimizing metamerism, in *Proc. IS&T/SID Eighth Color Imaging Conference: Color Science, Systems and Applications*, Scottsdale, AZ, November 2000, 34–38.
306. R. Vetter, C. Ward, and S. Shapiro, Using color and text in multimedia projections, *IEEE Multimedia,* 2(4), 46–54, 1995.
307. F. Viénot, Report on a fundamental chromaticity diagram with physiologically significant axes, in R. Chung and A. Rodrigues, Eds., *Proc. SPIE: The 9th Congress of the International Colour Association,* 4421, 565–570, 2001.
308. J. A. S. Viggiano, Modeling the color of multi-colored halftones, *TAGA Proc.,* 44–62, 1990.
309. P. L. Vora, J.E. Farrell, J. D. Tietz, and D. Brainard, Image capture: Simulation of sensor responses from hyperspectral images, *IEEE Trans. Image Proc.,* 10(2), 307–316, 2001.
310. P. L. Vora and H. J. Trussell, Measure of goodness of a set of color scanning filters, *J. Opt. Soc. Am. Abstr.,* 10(7), 1499–1508, 1993.
311. P. L. Vora and H. J. Trussell, Mathematical methods for the analysis of color scanning filters, *IEEE Trans. Image Proc.,* 6(2), 321–327, 1997.
312. P. L. Vora and H. J. Trussell, Mathematical methods for the design of color scanning filters, *IEEE Trans. Image Proc.,* 6(2), 312–320, 1997.
313. P. L. Vora, H. J. Trussell, and L. R. Iwan, A mathematical method for designing a set of colour scanning filters, in J. Bares, Ed., *Proc. SPIE: Color Hard Copy and Graphic Arts II,* 1912, February 1993, 322–332.
314. M. J. Vrhel, R. Gershon, and L. S. Iwan, Measurement and analysis of object reflectance spectra, *Color Res. Appl.,* 19(1), 4–9, 1994.
315. M. J. Vrhel and H. J. Trussell, Color correction using principal components, *Color Res. Appl.,* 17(5), 328–338, 1992.
316. M. J. Vrhel and H. J. Trussell, Filter considerations in color correction, *IEEE Trans. Image Proc.,* 3(2), 147–161, 1994.
317. M. J. Vrhel and H. J. Trussell, Optimal color filters in the presence of noise, *IEEE Trans. Image Proc.,* 4(6), 814–823, 1995.
318. E. Walowit, Ed., *Proc. SPIE: Device-Independent Color Imaging and Imaging Systems Integration,* 2170, February 1994.
319. B. A. Wandell, The synthesis and analysis of color images, *IEEE Trans. Pattern Anal. Mach. Intel.,* PAMI-9(1), January 1987, 2–13.

320. F. Werblin, A. Jacobs, and J. Teeters, The computational eye, *IEEE Spectrum,* 33(5), 30–37, 1996.
321. K. I. Werner, The flat panel's future, *IEEE Spectrum,* 30(11), 18–26, 1993.
322. G. West and M. Brill, Necessary and sufficient conditions for von Kries chromatic adaptation to give color constancy, *J. Math. Biol.,* 15(2), 249–258, 1982.
323. J. C. Whitaker, *Electronic Displays: Technology, Design, and Applications,* McGraw-Hill, New York, 1994.
324. M. Wolski, J. P. Allebach, C. A. Bouman, and E. Walowit, Optimization of sensor response functions for colorimetry of reflective and emissive objects, in E. Walowit, Ed., *Proc. SPIE: Device-Independent Color Imaging and Imaging Systems Integration,* 2170, February 1994, 209–219.
325. M. Wolski, J. P. Allebach, C. A. Bouman, and E. Walowit, Optimization of sensor response functions for colorimetry of reflective and emissive objects, *IEEE Trans. Image Proc.,* 5(3), 507–517, 1996.
326. M. Wolski, C. A. Bouman, J. P. Allebach, and E. Walowit, Optimization of sensor response functions for colorimetry of reflective and emissive objects, in *Proc. IEEE Int. Conf. Image Proc. 1995,* II, November 1995, 323–326.
327. S. L. Wright, R. W. Nywening, S. E. Millman, J. Larimer, J. Gille, and J. Luszcz, Image quality issues for height resolutions TFTLCDs, in *Proc. IS&T/SID Seventh Color Imaging Conference: Color Science, Systems and Applications*, Scottsdale, AZ, 100–105, 1999.
328. W. D. Wright, A re-determination of the trichromatic coefficients of the spectral colours, *Trans. Opt. Soc.,* 30(4), 141–164, 1928–29.
329. W. D. Wright, The sensitivity of the eye to small color differences, *Proc. Phys. Soc.,* 53(296), 93–112, 1941.
330. W. D. Wright, The basic concepts of colour order systems, *Color Res. Appl.,* 9(4), 229–233, 1984.
331. S. M. Wuerger, A. B. Watson, and A. Ahumada, Towards a spatio-chromatic standard observer for detection, in B. E. Rogowitz and T. N. Pappas, Eds., *Proc. SPIE: Human Vision and Electronic Imaging VII,* 4662, 159–171, 2002.
332. G. Wyszecki, Valenzmetrische Untersuchung des Zusammenhanges zwischen normaler und anomaler Trichromasie, *Die Farbe,* 2, 39–52, 1953.
333. G. Wyszecki and G. H. Felder, Color difference matches, *J. Opt. Soc. Am.,* 61(11), 1501–1513, 1971.
334. G. Wyszecki and G. H. Felder, New color matching ellipses, *J. Opt. Soc. Am.,* 61(9), 1135–1152, 1971.
335. G. Wyszecki and W. S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae,* 2nd ed., John Wiley & Sons, New York, 1982.
336. M. Xia, E. Saber, G. Sharma, and A. M. Tekalp, End-to-end color printer calibration by total least squares regression, *IEEE Trans. Image Proc.,* 8(5), May 700–716, 1999.
337. M. Yamaguchi et al., Color image reproduction based on the multispectral and multiprimary imaging: experimental evaluation, in R. Eschbach and G. G. Marcu, Eds., *Proc. SPIE: Color Imaging: Device Independent Color, Color Hard Copy, and Applications VII,* 4663, January 2002, 15–24.
338. L. A. Yoder, The TI digital light processing micromirror tech: putting it to work now, *Advanced Imaging,* 11(6), 43–46, 1996.
339. J. M. Younse, Mirrors on a chip, *IEEE Spectrum,* 30(11), November 1993, 27–31.
340. J. A. C. Yule and W. J. Neilsen [sic], The penetration of light into paper and its effect on halftone reproduction, in *TAGA Proc.,* 7–9 and 65–76, 1951.

341. J. A. C. Yule, *Principles of Color Reproduction, Applied to Photomechanical Reproduction, Color Photography, and the Ink, Paper, and Other Related Industries,* John Wiley & Sons, New York, 1967.
342. X. Zhang, E. Setiawan, and B. Wandell, Image distortion maps, in *Proc. IS&T/SID Fifth Color Imaging Conference: Color Science, Systems and Applications,* Scottsdale, AZ, November 1997, 17–20.
343. X. Zhang, D. A. Silverstein, J. E. Farrell, and B. A. Wandell, Application of a spatial extension to CIELAB, in *Proc. SID Int. Conference,* 1997.
344. X. Zhang, D. A. Silverstein, J. E. Farrell, and B. A. Wandell, Color image quality metric S-CIELAB and its application on halftone texture visibility, in *Proc. IEEE COMPCON,* 1997.
345. X. Zhang and B. A. Wandell, Spatial extension of CIELAB for digital color image reproduction, in *Proc. SID Int. Conf.,* 1996. Matlab code for the model is available at http://white.stanford.edu/ brian/scielab/.
346. M. D. Zmura, Color contrast gain control, in W. G. K. Backaus, R. Kliegl, and J. S. Werner, Eds., *Color Vision: Perspectives from Different Disciplines,* Walter de Gruyter, Berlin, 251–266, 1998.
347. J. C. Zwinkels, Colour-measuring instruments and their calibration, *Displays,* 16(4), 163–171, 1996.

chapter two

Visual psychophysics and color appearance

Garrett M. Johnson
Mark D. Fairchild
Rochester Institute of Technology

Contents

2.1 *Introduction*

What is color appearance, and how does it relate to digital color imaging? Color appearance is, as the name suggests, the study of how a given color stimulus is perceived by a human observer. While seemingly straightforward at first glance, color appearance is governed by the extraordinarily complex human visual system. How a stimulus appears is a function of many variables, ultimately including the spectral properties of the stimulus and the light source in which it is viewed; the size, shape, and spatial properties; and relationships of the stimulus, the background and surround, observer experience, and the adapted state of the observer.

Consider a relatively simple color imaging system that consists of a CRT computer display and a color printer. The desired goal of the system might be to have color images displayed on the CRT monitor that match the hard copy color images printed. It might be thought that using standard CIE

tristimulus colorimetry to assure that the XYZ values displayed by the monitor are exactly the same as those on the printed paper would be enough to assure a visual match. As it turns out, this tristimulus match between the monitor and paper would look very different to a human observer. This is because CIE colorimetry was designed with a very specific goal — that two simple stimuli that have identical tristimulus values match, for an average observer, under a single specified viewing condition. The above-mentioned color imaging system violates the assumptions that basic colorimetry requires. The two stimuli, in this case the CRT and print images, are complex stimuli viewed in wildly disparate conditions.

What is needed in the above situation is a method for ensuring that the *appearance* of the two images is identical. To do this, we first must understand what governs the appearance of a stimulus. This chapter focuses on just that problem. To fully understand how colors are perceived, it is important to understand the tools used to study color appearance.

In this chapter, we examine the terminology of color and color appearance. This includes appearance attributes such as hue, chroma, lightness, brightness, and saturation, as well as viewing condition attributes such as surround and background. We also examine several of the factors that influence color appearance and how they might cause basic tristimulus colorimetry to fail. Because color is ultimately the result of human perception, it is important to understand the tools used to quantify perception. Examples of these tools and techniques, known as visual psychophysics, are described.

Ultimately, we would like to be able to describe and predict the color appearance of complex stimuli under various viewing conditions. Toward this goal, there is the formulation of chromatic adaptation models and, ultimately, color appearance models. Several of these models and their historical formulation are described.

The study of color appearance is truly complex. This chapter barely scratches the surface of such a large and diverse field. The interested reader is encouraged to look at Fairchild's[1] more in-depth text on color appearance as well as the other references presented here. In addition, this topic can still be considered one of active research.

2.2 Terminology

In any field of study, it is important to have a common vocabulary so that knowledge and insight might be communicated accurately and precisely. In the study of color and color appearance, this vocabulary is often muddled, as terms such as *lightness* and *brightness* are often confused and casually interchanged by the average user. Why might this be more the case when discussing color as opposed to other subjects? Perhaps it is the very nature of color itself. Almost every person has experienced and discussed color, often at a very early age, though the means of discussion are often varied. Even in education, treatment of color is inconsistent. To the grade-school child, color might be made up of three primaries: red, blue, and yellow. The

printer is taught that the three primaries are cyan, magenta, and yellow, while the television engineer is taught that color is made up of red, green, and blue. In yet another manner, the physicist might be taught that color is made up of a certain portion of the electromagnetic spectrum. While all of these can be considered correct, they also can be considered incorrect.

In the field of color appearance, the *de facto* standard for vocabulary comes from the *International Lighting Vocabulary*, published by the CIE.[2] Hunt provides very useful insight into the need for a standardized vocabulary and also describes the work that led to publication of the CIE document.[3,4] To add to this mixture, there is also a relevant American Society for Testing and Materials (ASTM) document that describes appearance.[5] The definitions of terms presented below come directly from these important works.

2.2.1 Color

Perhaps some of the confusion in the field of color appearance stems from the very nature and definition of color itself. Few people, when asked, can give a precise definition of what exactly color is. It is almost impossible to do without using an example, as evident from the CIE definition.

> *Color***.** Attribute of visual perception consisting of any combination of chromatic and achromatic content. This attribute can be described by chromatic color names such as yellow, orange, brown, red, pink, green, blue, purple, etc., or by achromatic color names such as white, gray, black, etc., and qualified by bright, dim, light, dark, etc., or by combinations of such names.

This definition provides little satisfaction to the casual reader. To comfort those readers, it also provides little satisfaction to the scientists who study color. That the definition of *color* contains the word *color* makes for a circularity that can be confusing. The authors of this definition were well aware of this confusion and added a note that sums up the need for the study of color appearance.

> *Note*. Perceived color depends on the spectral distribution of the color stimulus, on the size, shape, structure, and surround of the stimulus area, on the state of adaptation of the observer's visual system, and on the observer's experience of the prevailing and similar situation of observations.

Many of the aspects described in this note will be discussed in much further detail later in this chapter.

Perhaps the most important information that is encompassed in this definition of color is the first sentence. Color is an attribute of visual perception. All terminology discussed in this section is similarly based on attributes of perception. That is to say, without the observer, there can be no discussion of color. The study of color appearance and color appearance models is an attempt to generate physically realizable measurements that correlate with these perceptual attributes.

2.2.2 Related and unrelated colors

The definition of *color* is further enhanced with the notion of related and unrelated colors. Though simple enough, these definitions are critical to gaining a full understanding of color appearance.

> *Related color.* Color perceived to belong to an area of object seen in relation to other colors.

> *Unrelated color.* Color perceived to belong to an area of object seen in isolation from other colors.

These definitions are rather straightforward. Related colors are viewed in relation to other color stimuli, while unrelated colors are viewed in isolation. Color stimuli are rarely viewed in complete isolation, so most color appearance models are designed to predict related colors. However, many color vision experiments that have been used to gain an understanding of the human visual system have been performed using simple unrelated color stimuli. It is important to understand the differences between these stimuli when trying to utilize models designed to predict one specific type of color.

Many color perceptions exist only for related or unrelated colors. One very interesting case is of the perceptions of colors such as brown and gray. These colors exist only as related colors. It is impossible to find an isolated brown or gray stimulus, as evidenced by the lack of a brown or gray light source. These lights would appear either orange or white when viewed in isolation. Likewise, all of the "relative" perceptions defined below only exist for related colors.

2.2.3 Hue

Hue is perhaps the easiest of the color terms to understand. Still, it is almost impossible to define hue without using examples. The CIE recognized this in its definition.

> *Hue.* Attribute of a visual sensation according to which an area appears to be similar to one of the perceived colors: red, yellow, green, and blue, or to a combination of two of them.

> *Achromatic Color.* Perceived color devoid of hue.

> *Chromatic Color.* Perceived color possessing a hue.

Hue is often described with a "hue circle," as shown in Figure 2.1. One important note of this description, and the definition given by the CIE, is the notion of unique hues; that is, red, yellow, green, and blue. These hues follow the opponent color theory first postulated by Hering in 1920.[6] Hering noted that certain hues were never perceived together. That is to say, there is no perception of a reddish-green, or a yellowish-blue. This formulated the fundamental notion that human color vision is encoded into red–green and blue-yellow channels. The interested reader is encouraged to read more

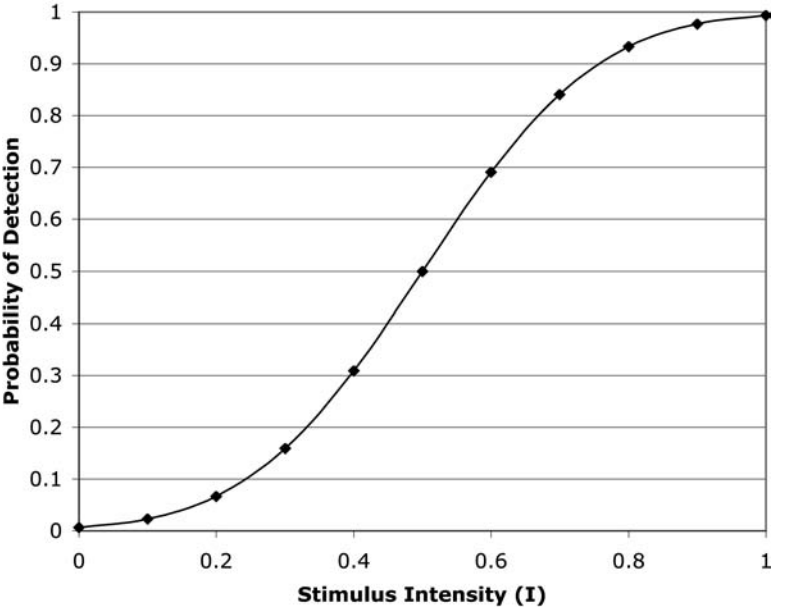


Figure 2.1 Example of a psychometric function.

thorough explanations as found in Kaiser and Boynton, Wandell, and Hurvich.[7–9]

The inclusion of the definitions for achromatic and chromatic colors is also important. Though often described as an interval hue circle, there is no natural meaning for a hue of "zero." Achromatic colors describe colors that are devoid of any hue information, but this definition does not extend to a meaningful interval hue scale. The meaning of different numerical scales will be described later in this chapter.

2.2.4 *Brightness and lightness*

The attributes of brightness and lightness are very often interchanged, despite the fact that they have very different definitions.

Brightness. Attribute of a visual sensation according to which an area appears to emit more or less light.

Lightness. The brightness of an area judged relative to the brightness of a similarly illuminated area that appears to be white or highly transmitting.

Note: Only related colors exhibit lightness.

Brightness refers to the absolute perception of the amount of light of a stimulus, while lightness can be thought of as the relative brightness. The

human visual system generally behaves as a lightness detector, which can perhaps be better described with an example.

A very simple example can be seen with a typical newspaper. This paper, when read indoors, would have a certain brightness and lightness. When viewed side by side with standard office paper, the newspaper often looks slightly gray, while the office paper appears white. When the newspaper and office paper are brought outdoors on a sunny summer day, they would then have much higher brightnesses. Yet the newspaper still appears darker than the office paper, as it has a lower lightness. The physical amount of light reflected from the newspaper might be more than a hundred times greater than the office paper was indoors, yet the relative amount of light reflected has not changed. Thus, the relative appearance between the two papers has not changed.

The above definitions include a note stating that only relative colors can exhibit lightness. This is the reason why there cannot be a gray light source. When viewed in isolation, the light source would be the brightest stimulus in the field of view and would thus appear white.

Further on in this chapter will be a discussion of color appearance models, which attempt to predict these appearance attributes. The various color appearance terms can get easily confused. Often, it is convenient to represent the relative terms with simple equations so as to gain a better understanding. Equation 2.1 shows the simple mathematical construct for lightness.

$$\text{Lightness} = \frac{\text{Brightness}}{\text{Brightness (white)}} \tag{2.1}$$

2.2.5 *Colorfulness and chroma*

The definitions of colorfulness and chroma are very similar to those of brightness and lightness, in the fact that colorfulness is an absolute perception, while chroma is relative.

> *Colorfulness*. Attribute of a visual sensation according to which the perceived color of an area appears to be more or less chromatic.
>
> *Note*. For a color stimulus of a given chromaticity and, in the case of related colors, of a given luminance factor, this attribute usually increases as the luminance is raised, except when the brightness is very high.
>
> *Chroma*. Colorfulness of an area judged as a proportion of the brightness of a similarly illuminated area that appears white or highly transmitting.
>
> *Note*. For given viewing conditions and at luminance levels within the range of photopic vision, a color stimulus perceived as a related color, of a given chromaticity, and from a surface having a given

luminance factor, exhibits approximately constant chroma for all levels of luminance except when the brightness is very high. In the same circumstances, at a given level of illuminance, if the luminance factor increases, the chroma usually increases.

Essentially, colorfulness describes the amount or intensity of the hue of a color stimulus. Similarly, chroma is to colorfulness as lightness is to brightness. This is also shown in Equation 2.2. Similarly to lightness, the human visual system generally behaves as a chroma detector. It is interesting that the appended notes attached to the above definitions are much longer than the definitions themselves. When the luminance of the viewing conditions increases, the chroma tends to remain constant as the brightness of a white stimulus is increasing as well. However, in this same situation, the colorfulness generally increases. This can be visualized by thinking of an outdoor scene. On a sunny day, everything looks very colorful, while on a cloudy day everything appears less colorful.

$$\text{Chroma} = \frac{\text{Colorfulness}}{\text{Brightness (white)}} \tag{2.2}$$

2.2.6 Saturation

Saturation is often confused with colorfulness and chroma, though it has its own unique definition.

Saturation. Colorfulness of an area judged in proportion to its brightness.

Note: For given viewing conditions and at luminance levels within the range of photopic vision, a color stimulus of a given chromaticity exhibits approximately constant saturation for all luminance levels, except when brightness is very high.

Whereas chroma is defined as the colorfulness of an area relative to the brightness of a similarly illuminated white stimulus, saturation is colorfulness relative to the brightness of itself. So, while only a related color can exhibit chroma, both related and unrelated colors can exhibit saturation.

The standard definition of saturation, as given above, can be seen in Equation 2.5. This definition can be supplemented with an alternate definition, which is used in some color appearance models. This definition says that saturation is the ratio of chroma and lightness. This is shown in Equation 2.3.

$$\text{Saturation} = \frac{\text{Chroma}}{\text{Lightness}} \tag{2.3}$$

By substituting the above definitions of lightness and chroma (Equations 2.1 and 2.2, respectively), we get Equation 2.4.

$$\text{Saturation} = \frac{\text{Colorfulness}}{\text{Brightness (white)}} \cdot \frac{\text{Brightness (white)}}{\text{Brightness}} \tag{2.4}$$

This equation can be simplified to the standard definition of saturation, as shown in Equation 2.5. It is important to note that, for unrelated colors, the ratio of chroma and lightness cannot be used to describe saturation, as those terms are only valid for related colors. When dealing with unrelated colors, Equation 2.5 must be used.

$$\text{Saturation} = \frac{\text{Colorfulness}}{\text{Brightness}} \tag{2.5}$$

2.2.7 *Digital color reproduction: brightness–colorfulness or lightness–chroma*

When dealing with color reproduction, often it is sufficient to represent color as trichromatic, as witnessed with the success of the CIE-based colorimetry. Colorimetry is valid only when dealing with color matches in identical viewing conditions. If the viewing conditions change, as when going from a CRT monitor to a print, colorimetry becomes insufficient. When this is the case, it becomes necessary to specify the actual color appearance. Complete specification requires five perceptual dimensions: brightness, lightness, colorfulness, chroma, and hue. It should be noted that the specification of saturation is not necessary. Saturation is redundant and can be inferred from the other percepts.

Many times, when designing imaging systems, it might appear that specifying all five color appearance attributes is also redundant. This is not the case, however, as was described by Nayatani et al.[10] In this article, Nayatani et al. describe the distinction between brightness–colorfulness (absolute) matches and lightness–chroma matches. For most imaging applications, it is often sufficient to attempt for a lightness–chroma match rather than the absolute brightness–colorfulness match. This can be illustrated by visualizing a common imaging system, such as consumer photography. Often, people photograph an outdoor scene in bright sunlight. The photograph is then printed and viewed in an indoor environment, at much lower luminance levels. In this case, it is physically impossible to achieve an absolute brightness–colorfulness match so that the measured energy coming off the print is the same as the original outdoor environment. This same situation can be easily reversed: if the original photograph was taken indoors and then reproduced and viewed outdoors. In this situation, while physically possible to reproduce the absolute attributes, it is undesirable. The reproduction would have to be unreasonably dark to match the absolute attributes of the indoor scene. For these cases, and for most general imaging applications, it is desirable to create a lightness–chroma match for these reproductions so that the relationship between objects in the scene is held constant.

2.3 Visual psychophysics

To gain an understanding of color, one of the foremost requirements is to have a basic understanding of the human visual system. Traditionally, the study of the human visual system generally falls into two categories: physiology and psychophysics. The study of the physiology of the human visual system involves examining the functionality of the receptors and neurons of the eye and the brain. This study is beyond the scope of this chapter, though there are several excellent texts on the subject.[7–9] Visual psychophysics is a technique for examining the relationship between physical measurements of a stimulus with the perception of that stimulus. More details of the experimental methods described in this chapter can be found in various texts, notably by Fairchild,[1] Bartleson and Grum,[11] Gescheider,[12] Torgeson,[13] Thurstone,[14] and Engeldrum.[15]

Physiology and psychophysics are not the only means used to study the human visual system. To fully understand the complicated nature of vision, one must combine the effort of many disciplines. This includes, but is not limited to, physics, optics, chemistry, genetics, biology, and anatomy. The remainder of this chapter, however, will focus on the use of psychophysics to study color appearance.

2.3.1 Definition of psychophysics

Psychophysics is the scientific study of the relationships between physically measured stimuli and the sensations and perceptions of those stimuli. *Psychophysics* can also be defined as the methodology used to study the abovementioned stimulus-sensation relationship. An example of this study might be the relationship between physical amounts of light (stimulus) and perceived brightness (perception). Psychophysics can be used to generate quantitative measurements of color sensation and perception, though those are often thought of as being very subjective. These measurements of perception, when produced from a carefully designed experiment, are just as objective as any other physical measurement (such as temperature). The difference between physical and psychophysical measurements tends to lie in the uncertainty of those measurements. Whereas a physical measuring device tends to have relatively small amounts of uncertainty, psychophysical experiments might have higher uncertainties. Care must be taken to understand and consider these uncertainties.

2.3.2 Psychophysical techniques

Many different experimental techniques can be used to measure perceptions of stimuli. For visual experiments studying images and color appearance, these tend to fall into two broad classes: threshold and scaling experiments. Many other types of experiments can be used, including categorization, recognition, and reaction time, though those will not be discussed here.

Threshold techniques include detection, discrimination, and matching experiments. They are designed to measure visual sensitivity to small changes in stimuli, or perceptual equality. An example of a detection or discrimination technique used in imaging science is for developing and testing image compression algorithms. An original image might be viewed with a compressed image to determine if the difference can be detected. An example of matching would be to have a person adjust the amount of compression of an image until it appears to match the original.

Scaling techniques are designed to produce a relationship between physical and perceptual magnitudes. The above-mentioned relationship between physical amounts of light and perceived brightness falls into this category. Another example might be the relationship between perceived image sharpness with measured spatial frequency information in the image.

2.3.3 *Hierarchy of scales*

When creating a scalar relationship between physical and perceptual magnitudes, it is important to consider the nature and properties of that scale. Various psychophysical techniques might produce different types of scales, each with different mathematical properties and utilities. It is very important to understand what mathematical operations are permitted, or vast misinterpretations can result. Four types of measurement scales will be defined, each with varying mathematical complexity and power. These scales are nominal, ordinal, interval, and ratio.

Nominal scales. These are the simplest form of numerical scales. Numbers are used as names for objects. An example of this type of scale would be the numbers on players on a sports team. The values of the numbers have no meaning other than to identify the different players. Any mathematical operation performed on this type of scale is arbitrary (for example, doubling every player's number has no meaning). In color appearance, a nominal scale can be given to color names, such as reds, greens, yellows, and blues. This scale can then be used for determining the category of a given color stimulus.

Ordinal scales. These scales have magnitudes of order associated with them. Objects can be ranked in ascending or descending order based on the magnitude of a certain trait. An example of this type of scale would be the Olympic medals, where gold, silver, and bronze medals are given out for first, second, and third place, respectively. It is easy to determine the order of the contestants, but any other relationship between them is unknown. For instance, did the gold medal long jumper jump twice as far as the silver medal jumper? Or was the difference between the gold and silver jump the same as the difference between the silver and bronze? A color appearance example of this type of scale might be the sorting of a series of paint chips in order of lightness. The resulting scale would reveal only that one paint chip was lighter than others, and there would be no information as to how much lighter. The only mathematical operation that is valid for an ordinal

scale is the greater-than/less-than operator. Any other operation should be considered arbitrary.

Interval scales. An interval scale is any scale that has equally spaced units, or intervals. For example, in this type of scale, if one sample is judged to be one unit away from an anchor, and a second sample is judged to also be one unit away, though in a different direction, the differences between the anchor and the first or second sample is still said to be perceptually equal. There is no meaningful zero in an interval scale, meaning the value of zero is arbitrary. A real-world example of this would be the Fahrenheit and Celsius temperature scales. The zero value in the Celsius scale is arbitrarily defined to be the freezing point of water, while in the Fahrenheit scale it is said to be 32° below the freezing point of water. Because the zero is arbitrary, it is impossible to perform multiplication and division on an interval scale. For example, we cannot say that 64°F is twice as warm as 32°. We can say that the temperature difference between 32° and 42° is the same as that between 52° and 62°. All of the mathematical operators that are valid for nominal and ordinal scales are also valid for interval scales. Interval scales, however, also allow for addition and subtraction.

Ratio scales. Ratio scales hold the most mathematical power of all the scales. They have all the properties of the previous three scales, with the addition of a meaningful zero point. The meaningful zero adds the ability to equate valid ratios. A real-world example of a ratio scale would be the meter scale for height and length. It should be obvious that zero means there is no magnitude of height. In this case, 8 m is indeed twice as long as 4 m and half as long as 16 m. Ratio scales also allow for the multiplication of constants without losing the meaning of the scale. An example of this would be converting between meters and centimeters, or meters and feet. In color imaging, it is often desired, yet impossible, to calculate a meaningful ratio scale. A hue scale is an excellent example of this. While it is relatively easy to calculate an interval scale of hue, it is difficult to determine the meaning of zero hue. Thus, zero hue is often arbitrarily assigned a location on the scale (e.g., red).

2.3.4 *Threshold and scaling: a historical perspective on Weber, Fechner, and Stevens*

To properly study the psychophysical techniques used in color imaging applications, it is often beneficial to begin with some history of the technique. Three pioneers of psychophysics who still are making their mark in color science today are Weber, Fechner, and S. Stevens.[16,17]

Weber began his work in the early nineteenth century, studying the perception of lifted weights. He asked subjects to lift a given weight and then added weight until the subjects were able to notice a difference between the new weight and the original. This experiment was repeated with many different starting weights. Weber noted that, as the starting weight increased, the amount of added weight necessary to produce a noticeable change also

increased. His experiments tended to show that, for a given starting weight, I, the change in weight necessary to elicit a perceptual difference, ΔI, followed a constant ratio $\Delta I/I$. This stimulus change is often referred to as a *just noticeable difference*, or a JND.

This simple relationship was found to hold approximately true for many different stimuli and has since become known as *Weber's law.* These findings turn out to be rather intuitive and are quite common in everyday life. For instance, often times when in a crowded places with loud music, people are forced to yell to be heard by others. When the music suddenly stops, the person yelling is instantly heard by everyone. The sound level coming from the person's mouth does not change but, because the background stimulus suddenly drops, the change necessary to be heard becomes much smaller. Other examples include the inability to see a candle in sunlight, though the candle appears bright when placed in a darkened room. This is an example of light adaptation and will be discussed further in this chapter. Weber's law helps explain these phenomena.

Later in the nineteenth century, Fechner proposed a method for extending Weber's law to create a scale of sensation.[16] Fechner theorized that a JND was a unit of sensation, and thus he could integrate JNDs to create an appropriate scale of sensation. Fechner attempted to create a transformation from a physical intensity scale (such as measured weight) to a perceived sensation scale (perceived heaviness) where each JND was of equal size for all perceptual magnitudes. Fechner adapted Weber's law and assumed that the ratio $\Delta I/I$ was held constant in the limit. By integrating over that equation, for all stimuli I, it is possible to calculate a metric that equates equal ratios on the physical scale with equal increments on the perceptual scale. This solution ends up being a simple logarithmic relationship, $S = k \log (I)$, where S is the perceived sensation, k is some constant, and I is the measured physical intensity. This solution became known as *Fechner's law.*

The logarithm expressed by Fechner's law represents a compressive nonlinear relationship between the input stimulus intensity and the corresponding perceptual sensation. The compressive nature essentially means that, as the stimulus intensity increases, the perceived sensitivity to the stimulus decreases. Going back to the person shouting in the loud room, because the intensity of the background is so high, the sensitivity to the sound decreases, and the person must shout to be heard. When the music suddenly stops, the sensitivity increases, and the person shouting can suddenly be heard by everyone.

Fechner's law relies on several fundamental assumptions. First, it assumes that Weber's law is indeed valid for all stimulus intensity (in the limit, $\Delta I/I$ is a constant). His other assumption is that JNDs are indeed a valid unit of sensation and that JNDs can be integrated to form a magnitude scale. While the general compressive trends described by Fechner's law are often valid for many perceptions, they often do not follow the exact logarithmic shape. Perhaps, because the two main assumptions often break down in real-world situations, so Fechner's law is not always accurate. Nevertheless, his

contributions to the field psychophysics and vision science are quite substantial.

Nearly 100 years later, S. Stevens performed a series of experiments testing the limits of Fechner's law. He used magnitude estimation experiments to derive relationships for over 30 different physical stimuli with their resulting sensations. It was found that most of the relationships formed straight lines when plotted on a log-sensation/log-intensity plot, rather than the logarithmic relation predicted by Fechner's law. The different perceptions did not all form lines of the same slope. When plotted in log–log space, straight lines indicate power functions in a linear space, where the slope indicates the exponent of the power function. From these plots, Stevens suggested that the relationships between physical stimuli and their corresponding perceptual scales could be defined as power functions, where the exponents vary for different perceptions. The general form of this is shown below.

$$S = kI^{\gamma} \tag{2.6}$$

where S = perception
k = experimental constant
γ = exponential power value

An exponent greater than 1 results in an expansive relationship; as the physical stimulus increases, the perception increases at a greater rate. This is often the case when the stimulus might result in danger, such as the perception of pain. An exponent less than 1 results in a compressive relationship such as that described by Fechner's law.

The power function relationship between physical and perceptual scales has become known as *Stevens' power law.* It has been used to model many perceptions in color imaging, such as the prediction of lightness in the CIELAB color space. Details on that will be explained later in the chapter.

Weber, Fechner, and Stevens formed the basis for many of the psychophysical techniques still used to develop and test color and appearance today. It is important to note the specific differences between Weber's goals and Fechner and Stevens' goals. In determining the amount of weight necessary to elicit a noticeable change in perceived weight, Weber was determining the threshold of detecting a change, or a just noticeable difference. Fechner and Stevens extended this to determine a scale of perceptual differences. These two techniques represent the main areas of psychophysical study for general color appearance.

2.3.5 *Psychophysical methods: threshold techniques*

Weber's weight experiment was a classical psychophysical threshold experiment. Threshold experiments are designed to determine the perceptible limits to a change in a stimulus, or the just noticeable differences (JND). Two

differing types of threshold JNDs can be calculated: absolute and difference. An absolute threshold determines the minimum amount of stimulus necessary to be detected. An example of this type of threshold might involve an observer in a blackened room trying to detect a small flashing light. The threshold would be determined from no stimulus (the blackened room) to some stimulus (the flashing light). Difference thresholds determine the smallest change detectable from a given stimulus. Weber calculated a difference threshold when he added more weight to an already existing amount of weight.

Three classical types of psychophysical techniques are used for determining thresholds. Over the years, many different experiments have been developed based on these types. One of the overall goals of any visual experiment should be simplicity. Simplicity often comes at a price, however. The three techniques will be presented here in order of simplicity, with their corresponding advantages and disadvantages also presented. The techniques are:

- Method of adjustment
- Method of limits
- Method of constant stimuli

2.3.5.1 Method of adjustment

The method of adjustment is the most straightforward method for determining observer thresholds for a given stimulus. It this technique, the observer has control over the magnitude of the stimulus itself. The observer must adjust the magnitude of the stimulus to reach a desired goal, or criterion. Example criterion might include adjusting a stimulus until it is just barely perceptible (for an absolute JND) or adjusting a stimulus until it is different from another (for a difference JND). The threshold is then determined by taking the average adjustment across several trials as well as across several observers. The standard deviation between a single observer and across several observers can also be taken, and it provides an indication of the variance and precision.

An example method of adjustment experiment for color imaging might be determining the level of image compression that can be applied before the observer notices a difference. For this type of experiment, an observer might sit at a computer screen that has two images on it, an original and a compressed image. The observer might have a slider that can be moved to increase and decrease the amount of compression on the image. The task would be to adjust the compression on the image until the subject just notices a difference between the original and the compressed version. This would be performed several times, and the average would be the compression threshold for that given observer and image. The threshold might be different depending on the starting value of the compressed image. If the image starts out uncompressed, and the observer must increase the compression until noticing a difference, you might get one value. If the image starts out very

obviously compressed, and the observer must decrease the compression until it is just barely noticeable it, you would probably get a different value.

The method of adjustment technique is advantageous in that it is fast and very easy to implement. It is also easy to calculate a threshold from the data it produces. There are several problems with this technique, though, as illustrated above by the different thresholds determined from the different starting points. This tends to show up as a bias, whereby if the observer starts from above the threshold (obvious compression), the threshold may be higher than if the observer starts from below (no compression). This bias might result from a change in observer criterion from one trial to another, or from adaptation to the starting stimulus. The criteria must be carefully explained and understood at the beginning of the experiment, yet it still might vary across different sessions or even across different trials. This results in less precise threshold data than obtained with other methods. Due to this lack of precision and ease of implementation of this technique, the method of adjustment is often used as a pilot experiment to generate starting values for some of the more complicated methods described below.

2.3.5.2 Method of limits

The method of limits provides more precise threshold data than the method of adjustment, with a slight increase in complexity. In this technique, the experimenter rather than the observer controls the presentation of the stimuli. The experimenter presents the stimuli at predefined discrete magnitudes. These magnitudes are presented in either a descending or ascending series. For a descending series, the stimulus is first presented well above threshold. The observer then must report, either verbally or through a response-recording device, whether the stimulus is detected. If the observer sees the stimulus (responds "yes"), then a new stimulus with a decreased intensity is presented. This is repeated until the observer can no longer see the stimulus.

For an ascending series, the first stimulus is presented such that it is definitely not detectable. The observer is asked to respond "yes" if the stimulus is seen, or "no" if not. If the observer responds "no," the stimulus intensity is increased. This is repeated until the observer can see the stimulus.

The threshold is determined to be the average of when the observer first detects the stimulus in the ascending series or does not detect the stimulus in a descending series. It is not uncommon for the two series to produce different thresholds. This might be caused by adaptation to the presenting stimulus or from expectation errors. Running both ascending and descending series for a given observer is a way to compensate for these errors. To further reduce the errors, it is possible to simultaneously run interleaved ascending and descending series.

Another issue with the method of limits is determining the discrete levels of stimulus intensity to present in the series. Because this is a threshold experiment, the only information comes from the "transitions," or where the stimulus is first detected or undetected. Essentially, all of the other stimuli

provide no information. Often, the method of adjustment is used to get a rough idea as to where the transition point occurs so as to minimize the "wasted" trials. There is also the same possibility of a change in observer criterion as there is in the method of adjustment. Because the observer ultimately must respond yes or no as to whether the stimulus is detected, the criterion can be changed for any given trial.

2.3.5.3 *Method of constant stimuli*

The method of constant stimuli attempts to overcome the observer variability by locking the observer criterion. This results in a more precise threshold number. In this method, the experimenter chooses a fixed number of stimuli at various intensity levels around threshold. The number of stimuli can vary, but it is typically between 5 and 7. The stimuli are then presented to the observer repeatedly, in a random order. For each trial, the observer must respond as to whether the stimulus is perceived. Over the course of the experiment, the frequency with which each stimulus level is detected is recorded. From these data, a "frequency of detection" function can be derived. This is often referred to as a *psychometric function*, which relates the probability of detection with stimulus intensity level. An example of the psychometric function is shown in Figure 2.1. From this function, it is possible to determine the threshold of detection as well as the uncertainty. Typically, the threshold is chosen to be the stimulus level that has a 50% probability of detection. The psychometric function can be determined individually for each observer, through multiple repetitions of the trials, as well as for a population of observers.

Generally, two types of constant stimuli experiments are run: yes–no and forced-choice. In yes–no experiments, observers are simply asked to respond *yes* if they detect a given stimulus or *no* if they do not. The psychometric function is then fit to the percentage of *yes* responses for each discrete stimulus level. An intensity that corresponds to 50% *yes* responses is taken to be the threshold point. This method can be extended to a pass–fail technique for determining visual tolerances. In this situation, a reference stimulus is presented, and observers either "pass" a stimulus that is less than the reference or "fail" one that is greater. This technique has been used to develop color difference equations, which will be described later in the chapter. For that case, the reference stimulus was a color pair of known difference, and the observers were asked to pass color pairs that had less of a difference and fail pairs that had a greater difference. These techniques can still suffer from changing observer criteria between trials.

A forced-choice experiment eliminates the observer criterion from the overall results. This is accomplished by presenting the observer with either spatial or temporal alternatives. For example, when attempting to determine the threshold of image compression, a pair of images is presented on a screen. The observer is then "forced" to choose which side of the screen displayed the compressed image. This is known as two-alternative forced-choice. Alternatively, the images could be presented in one of two time intervals. The

observer is then forced to choose in which interval the compressed image was presented. A psychometric function is then plotted using the percentage of correct responses against the stimulus intensity level. In a two-alternative forced-choice experiment, the psychometric function ranges between 50 and 100%, rather than 0 and 100% as in a yes–no experiment. That is due to the "forced" response nature of the experiment, whereby each observer must always choose an interval or location. If the stimulus intensity is too low to be detected, then the observer must make a guess. When two alternatives are available, the guessing rate is 50%. A threshold level is typically taken to be 75% correct. By forcing the observer to choose, the observer's criteria cannot influence the results.

The increased precision available from the method of constant stimuli comes at a price of complicated experimental design. Ideally, the discrete intensity levels need to be chosen so that the threshold falls in the middle of the range of intensities, and so that the lowest and highest levels fall close to 0% detected and 100% detected, respectively. To maximize this range, a pilot study is often necessary. This can be done using a small number of pilot subjects with a larger number of samples or by using another method such as the method of adjustment. To obtain an accurate psychometric function, it is also necessary to have many trials for each given intensity level. This can be accomplished by having a smaller number of observers do a large number of trials, or by having a larger number of observers perform fewer trials. Given the amount of time necessary to perform these experiments, it is often more desirable to have a larger number of observers. Another consideration is the seemingly arbitrary nature of selecting the threshold level. It has been suggested here to choose the 50% value of the psychometric function for a yes–no experiment and a 75% level for a forced-choice experiment. This threshold level can be calculated more precisely, but at the expense of losing the actual psychometric function itself. These techniques are known as up–down staircase procedures.

Staircase procedures combine a modified method of limits with a forced-choice experiment. They are designed to adaptively measure the threshold point on the psychometric function. An experiment begins with a stimulus of a given magnitude presented to an observer. This can be either a forced-choice presentation or a method of limits presentation. The observer is asked to respond to the presentation. A "yes" response, or a correct decision, will cause the magnitude of the next stimulus to be decreased. A "no" response, or an incorrect decision, will cause the magnitude of the next stimulus to be increased. In this manner, the staircase narrows in on the transition threshold. There are many variations and rules that can be used with these techniques. These rules determine the overall precision of the threshold. Further details can be found in psychophysical texts.[11–13]

2.3.5.4 Matching techniques

Matching techniques are generally similar to the method of adjustment, with only the goal being different. Whereas the method of adjustment is used to

determine the threshold level of a just noticeable difference, a matching experiment is used to determine when two stimuli are not perceptibly different. This technique has been used extensively in the color imaging community and is the technique used to generate the CIE XYZ system of colorimetry. In that situation, observers controlled the mixture of three light sources to match a separate monochromatic light source. An example of this is shown in Figure 2.2.

Matching techniques have also been used in the study of chromatic adaptation and color appearance. These techniques include asymmetric matching, where the stimuli are presented separately in disparate viewing conditions. An example of this would be viewing a reference color under daylight illumination and then attempting to match the color under incandescent illumination.

2.3.6 *Psychophysical methods: scaling techniques*

Threshold data can be useful when attempting to determine information such as color tolerances, or compression limits. Oftentimes the goal is to generate a scale of perception, rather than a single threshold. Scaling experiments are used to derive relationships between sensations and physical

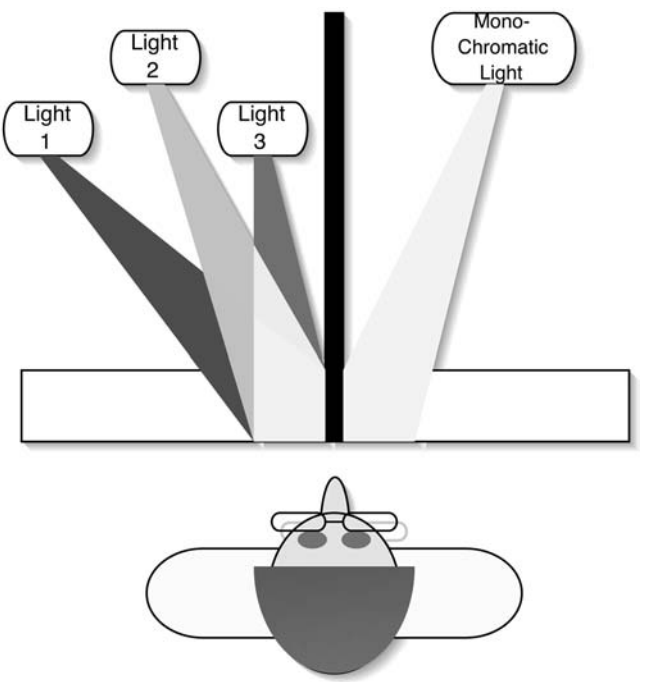


Figure 2.2 Example of a typical color matching experiment. The observer adjusts the three light sources on the left to match the single source on the right.

measurements of stimuli. Examples of scaling techniques were described briefly above in the discussion of Fechner's and Stevens' works. Several scaling techniques are used to generate these relationships. Depending on the dimensionality of the scale, different techniques are available. One-dimensional scaling is used when both the perceptual attribute and the physical measurement are one dimensional. Examples of this include scaling of lightness with luminance, where lightness is the perceptual attribute and luminance is the physical measurement. It is possible that an attribute being scaled actually consists of several distinct attributes, such as in the case of image preference. Image preference might result from several distinct variables, such as color fidelity, sharpness, and contrast. As long as the same criteria are used for each trial, one-dimensional scaling techniques can be used. Often, it is difficult to control the criteria, so more robust multidimensional scaling techniques should be used.

One-dimensional scaling techniques come in a variety of flavors. Some of the most common techniques for color imaging application are as follows:

- Rank order experiments
- Rating and category scales
- Partition scaling
- Magnitude and ratio estimation
- Paired comparison

Rank order experiments are generally simple to implement and perform. A series of stimuli are presented to an observer, who is asked to arrange the series in order of increasing or decreasing magnitudes. The magnitudes lie on the one-dimensional attribute that is being scaled. With enough observations, the data can be used to easily derive an ordinal scale of that particular attribute. Remember, the only mathematical operations that are valid for an ordinal scale are greater-than and less-than. Thus, the spacing between individual samples might not be equal. With enough samples and trials, it is possible to calculate an interval scale based on the law of comparative judgment, which will be described in more detail below. This involves many assumptions and simplifications and does not always produce accurate results. Interval scales should be generated at your own risk.

Rating and category scaling experiments allow for relatively simple determination of both ordinal and interval scales. Perhaps the simplest technique is the graphical rating scale. Observers are presented with a stimulus as well as a graphical scale with well-defined endpoints. The endpoints can be numerical, adjectival, or actual physical stimuli. For example, when attempting to scale chroma, the endpoints might say "no chroma" and "highest chroma imaginable," or simply 0 and 100. Observers are then asked to graphically mark where on the scale the current stimulus lies. The interval scale is then measured from the graphical scale. An example of scaling colorfulness using actual physical stimuli along with a graphical scale is shown in Figure 2.3.

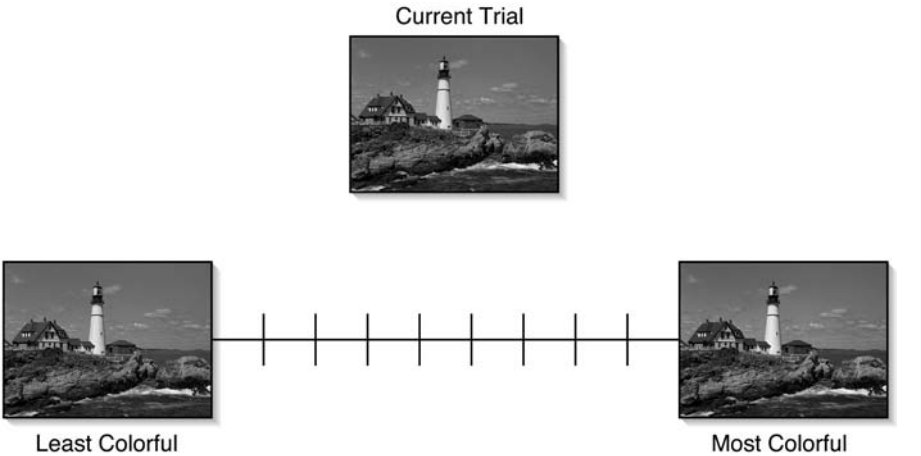


Figure 2.3 Scaling experiment using physical stimuli as endpoints.

Rating can also be performed without the benefit of the graphical scale, as an observer might be told the numerical endpoints verbally. When presented with a stimulus, the observer would then rate the perception by assigning a number between the endpoints. Another similar technique is called *category scaling*, or *adjectival rating*. This technique is useful when dealing with a large number of samples. An observer views a large sample population and is asked to separate the samples into predetermined categories, or adjectives. An oft-used example of categorical scaling is for sorting various hues into color names. An observer might be asked to place the samples into distinct color names categories such as red, green, yellow, blue, pink, orange, brown, black, gray, and white. This would result in a nominal scale of hue. More powerful scales are possible, such as an ordinal scale, if care is taken to select categories that can be considered equal intervals along the attribute being scaled. While this might be difficult when scaling hue, consider scaling colorfulness. The categories, or adjectives, given there might be "no colorfulness," "mildly colorful," "medium colorful," "very colorful," and "most colorful imaginable." If these categories are proximate enough that the categories into which the stimuli are placed are not the same for every person or observation, it is possible to generate an interval scale. This involves further statistical assumptions and the use of the law of categorical judgments.[13,14]

Partition scaling and fraction scaling are relatively straightforward experiments for the calculation of interval or ratio scales through a method of bisection. For example, in a partition scale experiment for image compression algorithms, an observer would be shown two images of different compression — images A and B. The observer would be asked to select a third image such that the difference between the third image and image A is the same as the difference between the third image and image B. Through successive bisection, as described above, a complete interval scale could be calculated.

When there is a distinct meaningful zero along the magnitude that is being scaled, it is possible to generate a ratio scale using these techniques. For instance, when scaling brightness, an observer might be presented with two spots of light and told to choose, or adjust, a third spot to be halfway between the first two spots. Alternatively, the subject might be flashed a spot of light and told to set a spot that is half as bright. Because there is a meaningful zero for brightness, no perceived light at all, this technique, through enough bisection, can create a ratio scale of brightness.

The above-mentioned fractional scaling can also be considered a form of *ratio estimation*. The easiest ratio estimation experiments are *magnitude estimation* or *production*. In a magnitude estimation experiment, an observer would be shown a stimulus and asked to assign a numerical value to that stimulus based on the magnitude of the sensation being scaled. In magnitude production, the observers are given a magnitude number, and they must adjust the stimulus so that it represents that perceptual magnitude. More complicated ratio experiments include the fractional brightness experiment described above, wherein an observer is asked to generate a stimulus that is half as bright as the previous stimulus. Another ratio estimation technique, given two or more stimuli, would be to have an observer state the perceived ratios between all the stimuli. For color imaging applications, where there is often no known meaningful zero, ratio estimation often proves too difficult.

Although it is often difficult to generate ratio scales in color imaging applications, interval scales can be generated with great success. A powerful technique for generating interval scales is *paired comparison*. Observers are presented with two stimuli and are asked to make ordinal judgments based on the pair. For example, given a pair of compressed images, an observer might be asked which image appears more compressed. This is valid only if the observers understand image compression and how compression artifacts might be manifested. It might be more desirable to have the observers choose which image is of higher quality, thus scaling quality as a function of compression. Alternatively, an observer might be presented with an original reference image and then asked which of the two compressed images looks most like the standard. This will create an interval scale of similarity, with regard to the image compression. Paired comparison experiments work well for a smaller number of samples and a well-defined ordinal criterion. To create an interval scale from these ordinal data, every possible pair of stimuli must be presented. That is, every stimulus must be compared with every other stimulus. For n stimuli, this leads to $n(n-1)/2$ experimental trials. Thus, the total number of trials increases very rapidly as the number of stimuli increases. Thurstone's laws of comparative judgment can then be applied to generate interval scales.[14] The law of comparative judgment has several underlying assumptions, among them that the perception of any stimulus results in a discriminal value on some psychological continuum and that, due to internal fluctuations, these discriminal processes result in a normal distribution of values. Assuming this normal distribution, then the average and standard deviation of the values relate directly with the average

and standard deviation of the perception itself. Thus, it is possible to convert the ordinal data derived from the paired comparison, using the power of the normal distribution, to a meaningful interval scale. The normal distribution also allows for the computation of statistically meaningful scales of similarity and differences between any given stimuli. There are several other simplifications and assumptions that can be made regarding the analysis using Thurstone's law. Bartleson and Grum[11] and Torgeson[13] give excellent details of all of these assumptions as well as worked-through examples.

2.4 *Viewing condition terminology*

Along with the standard color terminology given above, it is also important to have a sound understanding of the vocabulary used to describe the scene in which a stimulus is viewed. This scene is known as the *viewing field*, or more commonly as the *viewing conditions*. As you will see below, the viewing conditions can have a profound affect on the color perceptions. This section will define the common elements of a simplified viewing field, as shown in Figure 2.4. These elements are divided into four distinct components: stimulus, proximal field, background, and surround.

2.4.1 *Stimulus*

The stimulus is the color element of interest. In standard colorimetry, the stimulus is typically a small uniform color patch that subtends 2° of visual

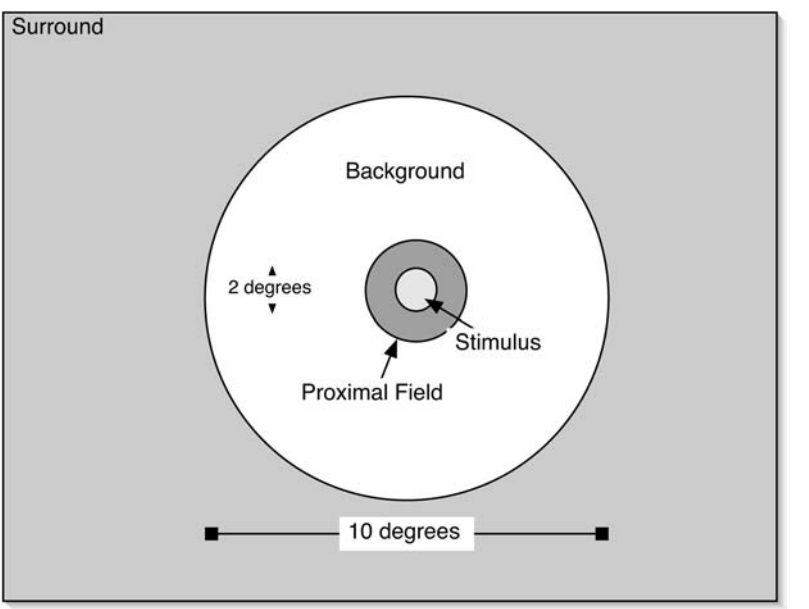


Figure 2.4 Specifications of the typical viewing field.

angle. The CIE has a separate system of colorimetry designed to handle larger color patches — the CIE 1964 supplemental standard observer. Most color appearance research has been performed using similar-sized uniform color patches. Ideally, the stimulus would be described by the full spectral representation. Often, this is difficult to do, if not impossible. When the spectral power distribution is unavailable, the stimulus is usually described using a standard device-independent space, such as CIE XYZ tristimulus values, or LMS cone responsivities.

For color imaging, the definition of the stimulus is somewhat blurred. Is the stimulus a single pixel, a region of pixels, or the entire image? While often more convenient to assume that the entire image is the stimulus, that might be an oversimplification. Currently, there is no universally correct definition of the stimulus for complex scenes. Therefore, when using images for research, care should be taken to fully describe the manner in which they are being used.

2.4.2 Proximal field

The proximal field is considered to be the immediate environment extending from the stimulus for about 2° in all directions. The proximal field can be useful for measuring local contrast phenomena such as spreading and crispening. These phenomena are described in detail later in this chapter. Ideally, the proximal field would also be described both spatially and with a full spectral power distribution. The question of defining the spatial proximal field becomes very difficult when dealing with digital color images. Should the proximal field for any given pixel be considered all of the neighboring pixels? In most real world applications, the proximal field is just assumed to be the same as the background.

2.4.3 Background

The background is defined to be the environment extending from the proximal field for approximately 10° in all directions. If there is no proximal field defined, then the background extends from the stimulus itself. Specification of the background is very important in color appearance, as it is necessary to model color appearance phenomena such as simultaneous contrast. Specifying the background with color patches is relatively straightforward. Specifying the background with color images suffers from the same problems as specifying the stimulus and the proximal field. For any given image pixel, the background actually consists of many of the neighboring pixels. Researchers generally use two different assumptions when determining the background for color imaging applications. The first is to assume that the entire image is the stimulus, so that the background is the area extending 10° from the image edge. Another assumption is that the background is constant and of some medium chromaticity and luminance, e.g., a neutral gray. Alternatively, the mean color of the image itself can be used as the

background. Because most imaging applications strive to reproduce images of constant spatial structure and size, many of these concerns disappear. Care must be taken when calculating color appearances across changes in image sizes, though. Braun and Fairchild describe the impact on some of these background decisions.[18]

2.4.4 Surround

The surround is considered to be anything outside of the background. For most practical applications, the surround is considered to be the entire room inhabited by the observer. Color appearance models tend to simplify the surround into a few distinct categories: dark, dim, and average. For instance, movie theaters are usually a dark surround, while televisions are viewed in a dim surround. More detailed discussion on the effect of the surround is given below.

2.4.5 Modes of viewing

Any changes in the above-mentioned viewing fields might result in a change in the color appearance of a stimulus. The following sections on color appearance phenomena explain some of these changes in detail. Other factors that cannot be readily explained by the simplified viewing field also have an effect on the perceived appearance of a stimulus. The perception of color is not adequately explained by the physics of light alone, as the human observer is the critical factor ultimately responsible for any sensation. The human visual system relies both upon sensory mechanisms, governed by biological and physical processes, as well as cognitive interpretations. These cognitive mechanisms are not fully understood, though we are able to recognize some behaviors. Perhaps one of the most important cognitive affects on color appearance is termed the *mode of appearance.* The mode of color appearance is a difficult concept to grasp at first and might be best described with an example.

Picture taking a walk outside on a clear winter night with only the full moon providing light. The snow on the ground probably will look very white, despite the fact that it is being illuminated almost entirely by the blue night sky. If you were to come across a house in the distance, the windows of the house might look bright orange. This orange light would be from the incandescent light bulbs found in most houses. If you were inside the same house, the light would not look nearly as orange and would most likely appear white. At the same time, the snow outside the window might look particularly blue. These are examples of changing modes of viewing, from object mode to aperture mode.

There are five modes of viewing that affect color appearance: illuminant, illumination, surface, volume, and film. These modes of viewing are described briefly below, though a more complete description can be found in *The Science of Color*, published by the Optical Society of America.[19]

The *Illuminant* mode of appearance is color appearance based on the perception of a self-luminous source of light. Because illuminant-color perceptions generally involve actual light sources, they are often the brightest perceptible color in the field of view. Examples of this are looking at a traffic light or an actual desktop light bulb. The immediate assumption that the brightest objects are actual light sources can lead to some interesting phenomena when non-illuminant objects in a scene appear much brighter than the surrounding scene. These objects might actually be perceived in an illuminant mode and are often described as *glowing*. Examples of an object appearing to glow might be when fluorescent objects are involved. Fluorescence is found in an object that absorbs energy (light) at one wavelength and emits the light at much longer wavelengths. Fluorescent objects are often referred to as "day-glow" objects, because they absorb light from nonvisible portions of the spectrum and emit light in the visible portions, thus appearing much brighter than the surrounding scene.

The *illumination* mode of appearance is similar to the illuminant mode, except that perceived color appearance is thought to be as a result of the illumination rather than properties of the objects themselves. Consider the traffic light example given above. Clearly, when looking at a traffic signal, there is no doubt that the red, yellow, or green color is being emitted from the light itself. Thus, the light is viewed in illuminant mode. Any pedestrians waiting for the light to turn might be bathed in red light and look quite red themselves. Generally, people do not assume that the pedestrians are very sick because of their red color. Instead, they recognize that the pedestrians are red because they are illuminated by the red traffic signal. The perceived color is a result of the prevailing illumination reflecting off the pedestrians' skin. Many clues are available to a typical observer of a scene when determining whether the color is a result of illumination. These clues include the color of the shadows, the color of the entire scene, as well as the color of the observer.

The perceived color of an observer or a pedestrian as described above is an example of the *surface* mode of appearance. In this mode, the color of a surface is perceived as belonging to the object itself. In the case of the pedestrians, the observer "knows" that the color of their skin and clothes belongs to them, and an observer is able to partly discount the color of the red traffic light. This is an example of "discounting the illuminant," which is described in further detail below. Any recognizable object provides an example of the surface mode of appearance. It requires both a physical surface and an illuminating light source.

The *volume* mode of appearance is similar to the surface mode, except the color is perceived to be "belonging" to a bulk or volume of a transparent substance. An example of volume mode appearance can be found in the perceived color of liquids, such as beer. The color of beer is not thought to be just on the surface but rather throughout the entire glass. As the beer is shaken up, forming a thick head, the air bubbles cause light to scatter, increasing the perceived lightness while decreasing the transparency. This

is an example of a volume color changing into a surface color. Volume color requires transparency as well as a three-dimensional shape and structure (the shape and structure of a glass of beer, for example).

The final mode of appearances, the *aperture or film mode*, encompasses all remaining modes of appearance. In the film mode, color is perceived as an aperture that has no connection with any object. In the moonlit walk example above, the orange window was perceived in an aperture mode of viewing. The observer did not believe that the window was glowing or that it was an actual light source. Rather, the window was perceived as an aperture. Any object can switch from surface mode to aperture mode if there is a switch in focus from the surface itself. This can be accomplished purposely by using an aperture screen or a lens system.

2.5 Color appearance phenomena

This section deals with examples of stimuli that do not follow the predictions of basic colorimetry. The CIE system of colorimetry was developed using a color-matching experiment that was similar to the magnitude adjustment experiments described above. Essentially, colorimetry states that, if two stimuli have identical tristimulus values, then those two stimuli will match each other for a given viewing condition. Colorimetry does not attempt to predict whether the colors will match if any aspect of the viewing condition changes. This section will illustrate several examples of where the color matches will indeed break down as various elements of the viewing conditions described in the previous section are changed. Among the changes in viewing condition are changes in illumination level, illumination color, surround, background, and viewing mode. The examples shown here illustrate the limitations of basic colorimetry and the need for advanced colorimetry, often called *color appearance modeling*. The foundations of most color appearance models stem from the study of these phenomena, so it is important to briefly review them here. The recognition and understanding of these color appearance phenomena are also important for a color imaging system designer, as many of these examples show up in everyday imaging applications. This section will describe several distinct forms of color appearance phenomena, including spatially structured, luminance, illuminant color, and surround effects.

2.5.1 Spatially structured phenomena

Perhaps the most easily recognized color appearance phenomenon is that of *simultaneous contrast*. Figure 2.5 illustrates an example of simultaneous contrast. The four small, gray patches are the same throughout the image. The two patches on the solid gray background look identical, while the patches on the white and black background look distinctly different. The patch on the white background looks darker, while the patch on the black background looks lighter. Simultaneous contrast causes the color of a stimulus to shift in color appearance when the color of the background changes. The change

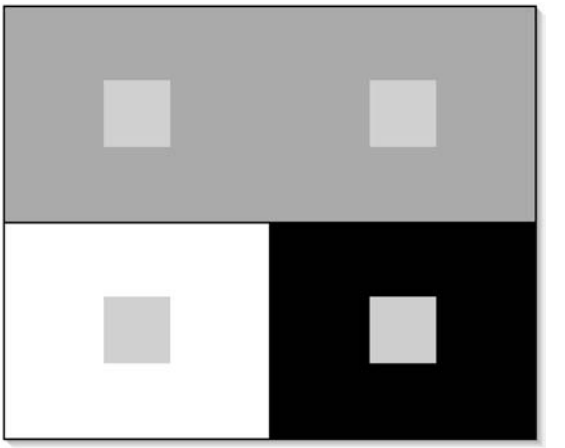


Figure 2.5 Example of simultaneous contrast. The four small, gray patches are identical.

in color of the stimulus tends to follow the opponent color theory of vision. That is why, in Figure 2.5, the patch on the white square looks darker, and the patch on the black square looks lighter. Simultaneous contrast can also be found with chromatic samples as well as achromatic. In those cases, following the opponent theory, a red background would tend to induce a green color shift, green would induce red, blue induces yellow, and yellow induces blue. Texts by Albers,[20] Fairchild,[1] Hurvich,[9] and Kaiser and Boynton[7] go into further detail regarding this phenomenon.

Figure 2.6 illustrates the complex spatial nature of simultaneous contrast. The centered ring in each of the circles is identical, as is the local contrast. The simultaneous contrast is shown to be much more apparent in the second circle pair. This suggests that spatial structure has a strong influence on simultaneous contrast. Robertson[21] and Shevell[22] present interesting examples as well as some models of this spatial relationship. As the spatial frequency of the stimulus increases, the contrast effect actually ceases and, in some cases, reverses.

At a sufficiently high spatial frequency, simultaneous contrast is replaced with *spreading*. With spreading, the color of a stimulus actually mixes with the color of the background. Recall that, with simultaneous contrast, the color of a stimulus took on the opposite color of the background. Often, it is hypothesized that spreading is caused by blurring of the light coming from the background with the light coming from the stimulus. While this might be true for very high-frequency stimuli, such as halftone dots, it does not fully explain the spreading phenomenon. Spreading can occur when the stimuli are very distinct from the background. An example of this can be seen in Figure 2.7.

Research is ongoing to understand the transition point between simultaneous contrast, spreading, and the overall effects of spatial frequency on

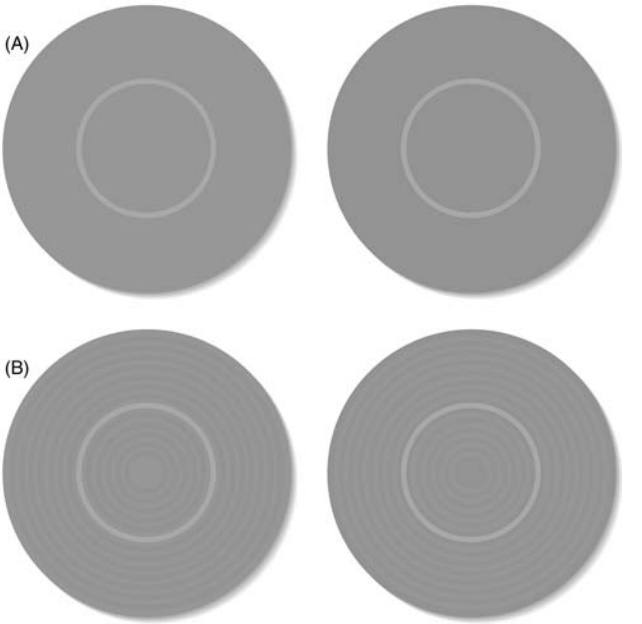


Figure 2.6 ***(See color insert following page 430)*** A spatially complex example of simultaneous contrast. The small inner rings are identical in size and color. The effect of the simultaneous contrast should be greater in the bottom pair (B).

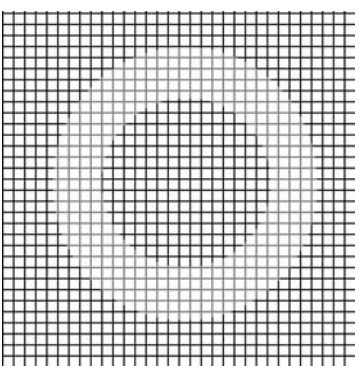


Figure 2.7 ***(See color insert)*** An example of spreading. There are only red and black lines, though a faint pink circle should be evident. (Adapted from Kuehni, R. G., *Color: An Introduction to Practice and Principles*, John Wiley & Sons, New York, 1997.)

color appearance.[22] Related, though more complex, phenomena include neon spreading and the watercolor effect. Neon spreading combines spreading with the perceptual attribute of transparency and is illustrated in Figure 2.8. Bressan[24] gives an excellent review of neon spreading. The watercolor effect, as seen in Figure 2.9, can also create strong spreading illusions.[25]

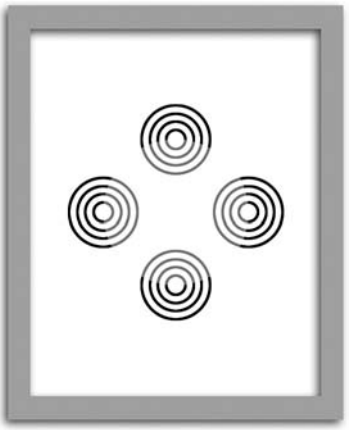


Figure 2.8 ***(See color insert)*** An example of neon spreading. There appears to be a transparent pink circle in the center of the figure.

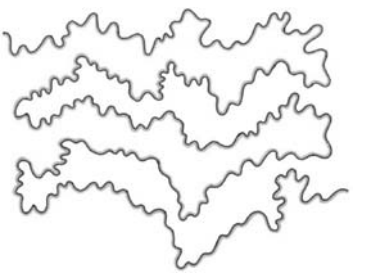


Figure 2.9 ***(See color insert)*** An example of the watercolor effect, where there appears to be surface colors caused by thin colored lines. Reproduced from Pinna et al.[25]

Simultaneous contrast can also give rise to an increase in perceived color difference between color stimuli. This effect is known as *crispening* and can be seen in Figure 2.10. Crispening causes an increase in perceived color difference when the background of the stimuli is close to the color of the stimuli. In Figure 2.11, the differences between the small gray patches are the same for all three backgrounds, but the difference looks greatest on the gray background. Similar effects can be seen for color patches as well. More details can be found in papers from Semmelroth[26] and, more recently, Moroney.[27]

2.5.2 *Luminance phenomena*

The above color appearance phenomena deal with color changes as a function of spatial structure and background. Profound color changes can also occur when the illumination stimuli are viewed under changes. This can include luminance level changes (dark to bright) or when the color of the

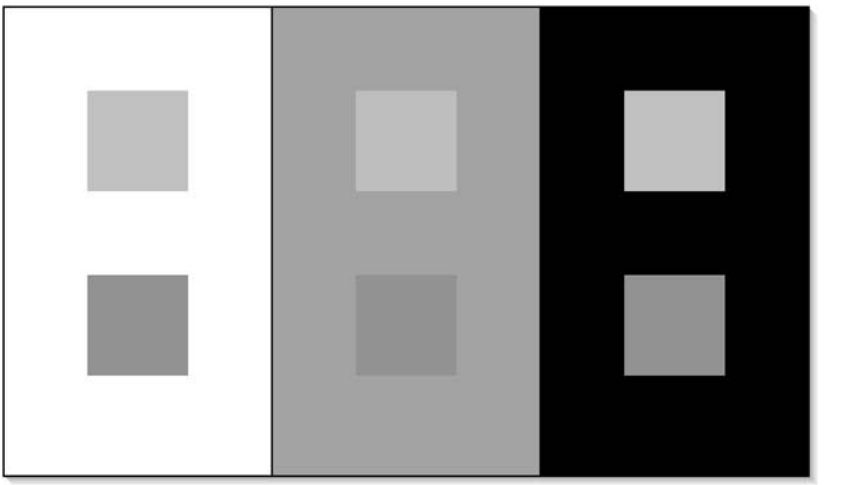


Figure 2.10 An example of lightness crispening. The color difference for the pairs of small squares are identical for each background, though they appear greatest on the gray background.

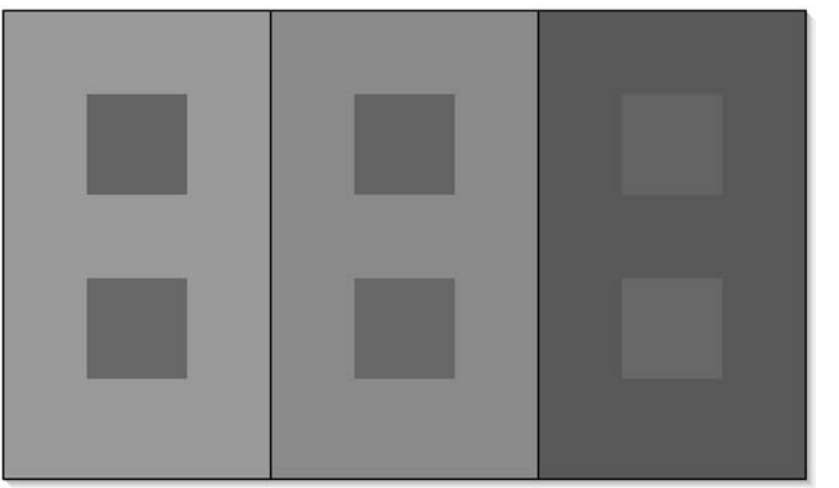


Figure 2.11 ***(See color insert)*** An example of chroma crispening. The color difference of the small pairs are identical but should look greatest on the background of most similar chroma (far right).

illumination changes. Luminance changes are very common in everyday life. The classic example is to think about a bright sunny day and a dark overcast day. Objects tend to appear very bright and colorful on sunny day and somewhat subdued on an overcast day. These occurrences can be well described by both the Hunt effect and the Stevens effect.

The *Hunt effect* states that, as the luminance of a given color increases, its perceived colorfulness also increases. This effect was first identified in a study by Hunt on the effects of light and dark adaptation on the perception of color.[28] Using a variation of a matching experiment called *haploscopic matching*, observers were given one viewing condition in their left eye and another in their right eye. Observers then used a method of adjustment technique to create matches on stimuli viewed in each eye. It was determined that, when one eye had a very low luminance level, it took much more colorimetric purity to match a stimulus viewed at a very high luminance level. This indicates that colorfulness is not independent of luminance level. Going back to the sunny day analogy, that partially explains why objects appear much more vivid, or colorful, when viewed in bright sunny environment. Scenes also appear much more contrasty when viewed in a bright environment.

This increase in contrast has been examined closely in a classic study by Stevens and Stevens.[29] This study showed that, as the luminance level increases, so too does the brightness contrast. This effect has been coined the *Stevens effect.* In this study, observers performed magnitude estimation experiments on brightness stimuli across many different luminance adapting conditions. This experiment has been described above in the discussion on Fechner's and Stevens' laws. The results showed that brightness tended to follow a power law relationship with luminance, thus forming the basis for Stevens' power law. However, this study also showed that the exponent of the power function changed as a function of adapting luminance level. Essentially, as the adapting luminance level increased, bright colors tended to look brighter, and darker colors tended to look darker. So, as the adapting luminance level increases, the rate of change between the brightness of the dark and light colors increases. This rate of change is often considered to be the contrast of the scene.

While the Stevens effect illustrates the change in brightness contrast with luminance level, what happens when there is a color change as well? Brightness is often erroneously assumed to be a function of luminance level alone. This is not the case, as is well illustrated by the Helmholtz–Kohlrausch effect. The Helmholtz–Kohlrausch effect shows that brightness also changes as a function of saturation. That is to say, as a stimulus becomes more saturated at constant luminance, its perceived brightness also increases. Another way to describe this effect is to say that a chromatic stimulus will appear brighter than an achromatic stimulus at the same luminance. If brightness were truly independent of chromaticity, then this effect would not exist. It is important to note that the Helmholtz–Kohlrausch effect is a function of hue angle as well. It is less noticeable for yellows than for purples, for instance. Essentially, this means that perceived brightness is actually a function of saturation and hue, and not just luminance. Fairchild and Pirrotta published a general review of the Helmholtz–Kohlrausch effect as well as some models for predicting the effect.[30]

Another interesting relationship between luminance level and chromatic colors is the Bezold–Brücke hue shift. This phenomenon relates the perceived

hue of monochromatic light sources with luminance level. It is often assumed that the hue of monochromatic light can be described completely by its wavelength. This is not the case, as the hue of a monochromatic light will shift as the luminance of the light changes. The amount of hue shift also changes, both in direction and magnitude as a function of hue. Experimental results regarding the Bezold–Brücke hue shift can be found in work published by Purdy.[31] One important consideration for these hue shifts is that all the experimental data were obtained using unrelated colors. Recall that unrelated colors are stimuli viewed in complete isolation. Unrelated colors occur very rarely in everyday life. Hunt published a report indicating that the Bezold–Brücke hue shift disappears for related colors.[32] This must be taken into consideration when creating a model to predict color appearance.

2.5.3 *Hue phenomena*

We have seen above how luminance changes can cause large shifts in the appearance of colored stimuli. This section examines two phenomena that result from changing the hue of the viewing conditions. These hue changes are less common than luminance changes, and they often are not very perceptible. They are included here because many models of color appearance models are capable of compensating for these effects.

The Bezold–Brücke hue shift illustrated that the wavelength of monochromatic light sources is not a good indicator of perceived hue. As luminance levels change, the perceived hue can also change. Another similar effect is the *Abney effect*. The Abney effect simply states that adding "white" light to a monochromatic light does not preserve constant hue. Another way of expressing this is to say that straight lines in a chromaticity diagram, radiating from the chromaticity of the white point to the spectral locus, are not lines of constant hue. Unlike the Bezold–Brücke hue shift, this effect is valid for related colors as well as unrelated colors.

Another interesting, though difficult to reproduce, phenomenon involving monochromatic illumination is the *Helson–Judd effect*.[33] This effect describes that nonselective (gray) stimuli viewed under highly chromatic illumination take on the hue of the light source if they are lighter than the background, and they take on the complementary hue if they are darker than the background. So, a dark gray sample viewed on a medium gray background under red illumination will look somewhat green, while a light gray sample would appear pinkish. This effect almost never occurs in common practice and is very difficult to reproduce in a laboratory setting. Nevertheless, some color appearance models take this into account. More details on this effect can be found in Fairchild[1] and Mori et al.[34]

2.5.4 *Surround phenomena*

The Stevens effect demonstrated that contrast for simple patches increased as a function of adapting luminance. Around the same time, Bartleson and

Breneman were studying the effects of luminance level and surround on complex stimuli, namely images.[35] They were able to generate results similar to those of Stevens and Stevens in regard to changes in luminance level. More interestingly, they noticed interesting results regarding the change in the relative luminance of the image surround. Recall that the surround is considered to be the field outside of the background or, in practical situations, the entire viewing room. Bartleson and Breneman determined that perceived contrast in images increased as the luminance of the surround increased. That is to say, when an image is viewed in a dark surround, the black colors look lighter while the light colors remain relatively constant. As the surround luminance increases, the blacks begin to look darker, causing overall image contrast to increase.

These results modeled phenomena that were already taken into account in the photographic world. Traditionally, for optimal tone reproduction, photographic transparencies designed for viewing in a darkened room were reproduced with a much higher contrast than those designed for viewing as a print in a bright room. Hunt[36] and Fairchild[37] provide more in-depth analysis of the history and prediction of optimal tone reproduction for complex images. In their original publication, Bartleson and Breneman published equations that predicted their results well. These equations were simplified later to create equations for calculating optimal tone reproduction.[38] Such equations have been adapted and are included in many models of color appearance.[1]

Surround compensation can play a key part in the design and implementation of a color imaging system. For instance, in designing a scanner to convert movie film into video for display on a television, one must understand the effects that surround will have on the final output image. Television is typically viewed in a lighter surround than a darkened movie theater. If the scanner does not take this change in surround into account, it is possible for the video to appear to have a much higher perceived contrast than the original film.

2.5.5 Color constancy and discounting the illuminant

Illumination can vary dramatically throughout many different environments. This includes both the physical amount of illumination and the color of the illumination. Several of the examples above illustrate how these changes in illumination can cause the appearance of colors to change drastically. At the same time, most people will readily acknowledge that the colors of objects do not change when moving from one viewing condition to another. A red apple will look red when viewed under bright outdoor illumination as well as when viewed inside at relatively dark incandescent illumination. This is the effect known as *color constancy*. One of the mechanisms for color constancy is chromatic adaptation, which is described in much further detail below. Suffice it for now to know that chromatic adaptation is a result of sensory adaptation as well as cognitive behavior. The

cognitive ability of an observer to interpret the color of an object based on the illuminated viewing environment is known as *discounting the illuminant*. Essentially, this is the mechanism that allows for observers to "know" that the red apple is still red, despite potentially large changes in the color of the illuminant. Color constancy is an area of active research. The publications by Jameson and Hurvich[39] as well as Fairchild[1] provide good starting points for the researcher interested in the study of color constancy.

2.6 Chromatic adaptation

The human visual system is capable of functioning across vast changes in viewing conditions while providing relatively stable perceptions. The mechanism that allows the visual system to do this is known as *adaptation*. Adaptation allows the general sensitivity to any given stimulus to change based on the conditions of the stimulus itself. Three types of adaptation are important for modeling vision and color imaging: light, dark, and chromatic. Light and dark adaptation describe the human visual system's capability of functioning across large changes in luminance levels. These changes need to be considered when building color imaging systems that are designed to work across wide luminance ranges, though those types of situations are relatively rare. Chromatic adaptation is the ability of the human visual system to adjust to changes in the *color* of illumination.

The previous section described many color appearance phenomena, or examples where basic tristimulus colorimetry fails. Several of these examples represent changes in luminance level, such as the Hunt and Stevens effects. Many of the other phenomena described above can be considered second-order effects, as the situations in which they occur happen relatively infrequently. Chromatic adaptation, and the similar concept of color constancy, are perhaps the most important of the color appearance phenomena. This section discusses the theory of chromatic adaptation and some of the mechanisms that enable adaptation. This section also describes some computational models of chromatic adaptation and how those models can be used to calculate color appearance matches across different viewing conditions. These matches are important when designing color imaging systems that are capable of reproducing colors for view in various conditions.

2.6.1 Light and dark adaptation

Light adaptation is the decrease in visual sensitivity as a function of the overall amount of illumination. Essentially, the more light illuminating a scene, the less sensitive the human visual system becomes to light. This is a very common occurrence. Imagine going to an afternoon cinema matinee. When leaving the darkened theatre into the sunny afternoon light, your visual system is often shocked — sometimes even to the point of physical pain. It is very difficult to see anything for a few moments, and then your visual system adjusts so that you can see objects normally. *Dark* adaptation

is the opposite; the human visual system becomes more sensitive to light as the overall amount of illumination decreases. This can be thought of as walking from the sunny afternoon light into a darkened theatre and struggling to find your seat. After several minutes, objects become recognizable as your visual system adapts.

Light and dark adaptation, though very similar, function at different speeds. The speed of adaptation is often referred to as the *time-course for full adaptation.* Light adaptation works at a much faster rate than dark adaptation. Consider the movie theatre discussion above. When leaving the theatre to go outside, it is somewhat painful for several seconds, and then vision returns to normal. Dark adaptation can take several minutes before objects become noticeable. This indicates the mechanisms of dark adaptation are much more gradual than those of light adaptation.

So what are the physiological mechanisms that enable light and dark adaptation? One mechanism has been reproduced almost identically in photographic camera systems. This is the dilation and constriction of the pupil in the eye. For many years, cameras have had an aperture control that enables the photographer to adjust the amount of light that enters the lens. The human eye works in a similar manner. In ordinary viewing situations, the pupil can range in diameter from about 3 to 7 mm. From these different diameters, we can conclude that the pupil can account for up to a 5× change in luminance level. Considering that the range in luminance levels from sunlight to starlight can differ upwards of 10 orders of magnitude, clearly, the pupil dilation and contraction cannot be the only mechanism of adaptation. Other mechanisms include the transition from cones to rods, and vice-versa. In the human retina, there are two distinct types of photoreceptors: rods and cones. Rods are more sensitive to light and are responsible for vision at low luminance levels. Cones are less sensitive to light and are responsible for color vision at higher luminance levels. The transition from cones to rods can account for additional levels of adaptation. Additionally, this transition can explain the difference in the time-course of adaptation between light- and dark-adaptation mechanisms. The cones respond relatively quickly to increased levels of illumination, while the rods respond more slowly to decreased levels. Other mechanisms can account for light and dark adaptation, including receptor gain control, where the photoreceptors themselves become less sensitive to light at increased luminance levels. Receptor gain control is perhaps the most important sensory mechanism for chromatic adaptation, and it will be revisited.

2.6.2 *Chromatic adaptation*

Chromatic adaptation refers to the human visual system's ability to adjust to the color of overall illumination rather than the absolute levels of the illumination. This is perhaps best explained with a common example. Consider a white object such as a piece of paper. This paper can be viewed under a variety of light sources such as daylight, incandescent, and fluorescent.

Despite the large change in the color of these sources (ranging from blue to orange), the paper will always retain an approximate white appearance. Chromatic adaptation is often thought to be a result of independent gain control mechanisms on the three types of cone photoreceptors, as illustrated in Figure 2.12. This is similar to the receptor gain control functions of light and dark adaptation, though those can also be explained with a single gain control function for all photoreceptors.

While this is certainly a valid hypothesis, there is no evidence that the gain control mechanisms do not also occur at other stages of visual processing. The theory of independent photoreceptor gain control was first published 100 years ago in a seminal paper by von Kries.[40,41] In that paper, translated by MacAdam, he wrote:

> …the individual components present in the organ of vision are completely independent of one another, and each is fatigued or adapted exclusively according to its own function.[40,41]

This insight, though now we know it is not entirely correct, provided an excellent starting point on the theory of chromatic adaptation.

The receptor gain control idea of chromatic adaptation is very similar in principle to an automatic white balance in a digital camera or camcorder. Those devices adjust the sensitivities of their detectors such that the "brightest" object in the scene appears white. This is accomplished by normalizing

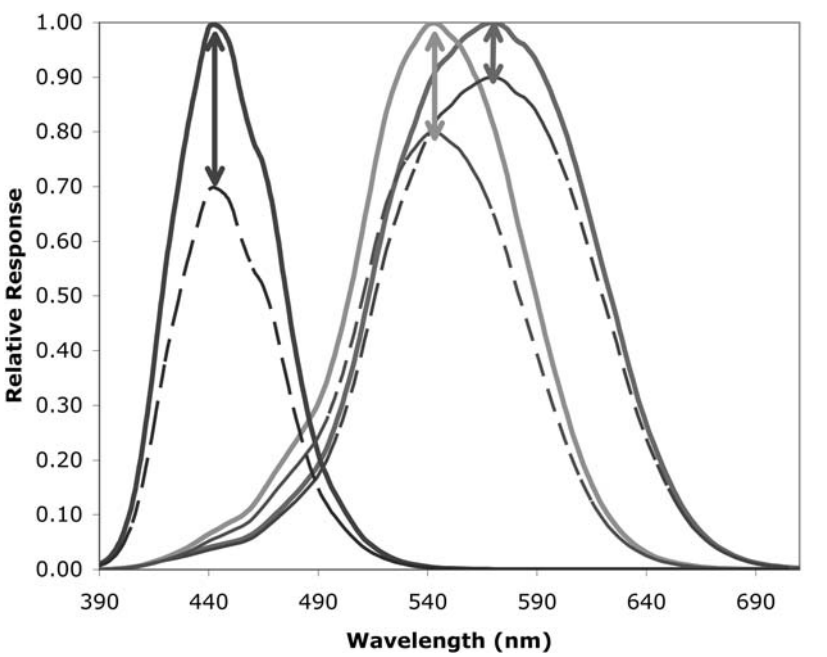


Figure 2.12 Iconic concept of independent cone gain control.

all the detectors with the strongest detector signal. This type of adaptation is classified as a *sensory* mechanism. A sensory mechanism is a mechanism that responds automatically to the stimulus energy. If chromatic adaptation were entirely a sensory mechanism, it would be much easier to understand and model. Unfortunately (at least from a modeling standpoint), chromatic adaptation is a combination of sensory and *cognitive* mechanisms. A cognitive mechanism responds to a stimulus based on an observer's knowledge of scene content.

Some cognitive mechanisms have been discussed in previous sections. Examples include *discounting the illuminant* and *color constancy.* Another interesting cognitive mechanism is *memory color.* Memory color is the phenomenon that recognizable objects often have a "known" color associated with them. Typical memory colors might be green grass, blue sky, skin tones, and the red apple example given above. Figure 2.13 illustrates the idea of cognitive mechanisms, and perhaps memory color, for a yellow banana. The image on the left of Figure 2.13 has a green filter placed over the entire image, while the image on the right has the filter only on the banana. The banana retains its yellow color when the entire image is filtered, while the identically colored banana looks greenish when it is the only object filtered.

When asked to produce memory colors in an experiment, using techniques such as the method of adjustment, observers are generally able to perform that task with relative ease. An interesting note is that memory colors often are remembered differently with respect to the actual object. For instance, when asked to produce a grass green, observers typically make a green that is much more saturated than actual grass.[42] Perhaps this is an indication of observer preference blending into memory color. Cognitive mechanisms of color appearance are discussed in much greater detail in works by Evans,[43] Jameson and Hurvich,[39] Davidoff,[44] and Fairchild.[1]

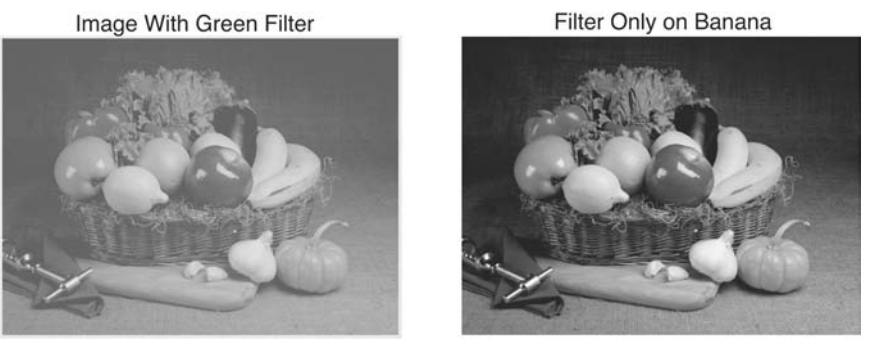


Figure 2.13 (See color insert) An example of cognitive mechanisms of chromatic adaptation. The image on the left is covered with a green filter. The image on the right has the same green filter placed only on the banana. The color of the banana is the same for both images, though it retains its yellow appearance in the left image and looks green in the right.

2.6.3 Chromatic adaptation models

Models of chromatic adaptation are the first step toward the creation of a color appearance model. Chromatic adaptation models extend the function of basic tristimulus colorimetry. Basic colorimetry was designed to predict appearance matches between stimuli within a single, constant viewing condition. The color appearance phenomena described in the sections above illustrated areas where basic colorimetry fails. Chromatic adaptation enables visual matches to persist through wide ranges of viewing conditions. Two stimuli that are viewed in different conditions, yet appear to match, are called *corresponding colors.* For example, one stimulus might be viewed under daylight simulators, while another is viewed under a tungsten light bulb. The two stimuli might have different XYZ tristimulus values but, because of chromatic adaptation to the illuminating light sources, they might appear to match.

Basic colorimetry is not designed to predict matches across different viewing conditions. To predict these matches, we need a model of chromatic adaptation. The general form of a chromatic adaptation was first described by von Kries, as discussed above.[40] He described a simple hypothesis for a model of chromatic adaptation based on cone photoreceptor normalization. There are two general misconceptions regarding von Kries' ideas for chromatic adaptation. Although many chromatic adaptation models claim to utilize a von Kries transformation, often called a *von Kries coefficient* or *proportionality law,* the equations used in these types of models were never actually proposed by von Kries. Rather, he simply proposed his idea for independent cone adaptation. This idea was meant to serve as an interim solution or a stepping stone for more advanced research. Little did he know that, 100 years later, his simple hypothesis would still be in widespread use.

2.6.4 von Kries model

Although von Kries himself did not formulate equations for chromatic adaptation, his hypothesis has been used to create a simple chromatic adaptation model. Many chromatic adaptation models are designed to work in conjunction with CIE colorimetry. The hypothesis laid out by von Kries suggested that the cone photoreceptors adapted independently of one another. To model this in a meaningful physiological manner, it is necessary to transform from CIE XYZ tristimulus values into LMS cone responses (sometimes referred to as RGB or ργβ responses). The LMS cone responses can be calculated fairly accurately using a linear transform of CIE tristimulus values. An example transformation, referred to as the Hunt–Pointer–Estevez transformation (normalized to illuminant D65), is described in Equation 2.7.[45]

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.4002 & 0.7076 & -0.0808 \\ -0.2263 & 1.1653 & 0.0457 \\ 0.0 & 0.0 & 0.9182 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2.7}$$

These LMS cone responsivities are then used in a modern interpretation of a chromatic adaptation model, known as a coefficient model. This interpretation is shown in Equations 2.8 through 2.10.

$$L_{adapted} = a_L \cdot L \tag{2.8}$$

$$M_{adapted} = a_M \cdot M \tag{2.9}$$

$$S_{adapted} = a_S \cdot S \tag{2.10}$$

In these equations, L, M, and S represent the initial cone responses to a given stimulus, and $LMS_{adapted}$ are the post-adaptation cone signals. To obtain the adapted cone signals, each LMS response is scaled using the independent gain control coefficients: α_L, α_M, and α_S. How these gain control coefficients are calculated is the key aspect to most chromatic adaptation models. For the typical von Kries model, those coefficients are described to be the inverse of the maximum LMS response in the scene. The maximum LMS response is typically the scene white, so a von Kries adaptation is often referred to as a *white-point normalization.* Equations 2.11 through 2.13 illustrate the idea of a white-point adaptation.

$$a_L = 1/L_{max} \quad \text{or} \quad a_L = 1/L_{white} \tag{2.11}$$

$$a_M = 1/M_{max} \quad \text{or} \quad a_M = 1/M_{white} \tag{2.12}$$

$$a_S = 1/S_{max} \quad \text{or} \quad a_S = 1/S_{white} \tag{2.13}$$

Often, it is convenient to express the chromatic adaptation model as a linear matrix transform. This is especially useful for concatenating transforms as well as when programming models. The above interpretation of the von Kries type of chromatic adaptation model is shown in Equation 2.14.

$$\begin{bmatrix} L_{adapted} \\ M_{adapted} \\ S_{adapted} \end{bmatrix} = \begin{bmatrix} 1/L_{white} & 0.0 & 0.0 \\ 0.0 & 1/M_{white} & 0.0 \\ 0.0 & 0.0 & 1/S_{white} \end{bmatrix} \cdot \begin{bmatrix} L \\ M \\ S \end{bmatrix} \tag{2.14}$$

By expressing the chromatic adaptation transform as a matrix transformation, we can generate adapted CIE XYZ tristimulus values with a single 3×3 transformation. This is shown in Equation 2.15.

$$\begin{bmatrix} X_{adapted} \\ Y_{adapted} \\ Z_{adapted} \end{bmatrix} = \mathbf{M}^{-1} \cdot \begin{bmatrix} 1/L_{white} & 0.0 & 0.0 \\ 0.0 & 1/M_{white} & 0.0 \\ 0.0 & 0.0 & 1/S_{white} \end{bmatrix} \cdot \mathbf{M} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2.15}$$

M and $\mathbf{M}^{-1}$ represent the Hunt–Pointer–Estevez transformation and inverse transformation, respectively, as illustrated in Equation 2.7. While it is useful to obtain the adapted CIE XYZ tristimulus values for a given stimulus, it is often more useful to obtain *corresponding colors* data. Recall that corresponding colors are two stimuli that appear to match when viewed under disparate conditions. A model that can calculate the tristimulus values necessary to obtain this perceptual match across different viewing conditions is known as a *chromatic adaptation transform,* or CAT.

2.6.5 *von Kries transform*

Once a chromatic adaptation model is available, it is very easy to extend it with the ability to "transform" CIE XYZ tristimulus values from one viewing condition to another. The general form of this transformation is shown in Equations 2.16 through 2.18.

$$L_2 = 2L_1 / L_{white\,2} \tag{2.16}$$

$$M_2 = (M_1 / M_{white}) \cdot M_{white\,2} \tag{2.17}$$

$$S_2 = (S_1 / S_{white}) \cdot S_{white\,2} \tag{2.18}$$

L_2, M_2, and S_2 are the predicted cone responses of the perceptual match for the original LMS responses, though under the second viewing conditions. L_{white}, M_{white}, and S_{white} are the cone responses of the white point in the original viewing condition, while L_{white2}, M_{white2}, and S_{white2} are cone responses of the white point in the new viewing conditions. These equations are essentially calculating the post-adaptation signals from the first viewing condition, designated LMS, and setting those signals equal to the post-adaptation signal from the second viewing condition. The chromatic adaptation model is then inverted to calculate the pre-adaptation response necessary to elicit that equal signal. The corresponding CIE XYZ tristimulus values can be found by concatenating Equation 2.15 with Equations 2.16 through 2.18. This is illustrated in Equation 2.19.

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} = \mathbf{M}^{-1} \cdot \begin{bmatrix} L_{white\,2} & & \\ & M_{white\,2} & \\ & & S_{white\,2} \end{bmatrix} \cdot \begin{bmatrix} 1/L_{white} & 0.0 & 0.0 \\ 0.0 & 1/M_{white} & 0.0 \\ 0.0 & 0.0 & 1/S_{white} \end{bmatrix} \cdot \mathbf{M} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2.19}$$

2.6.6 *Nayatani's model*

The von Kries model of chromatic adaptation is a relatively straightforward linear scaling of fundamental cone responsivities. This model was enhanced by Nayatani et al. to include a nonlinear term in addition to the linear gain

control.[46,47] The nonlinear model was extended from a nonlinear model first proposed by MacAdam.[48] The Nayatani model is essentially a von Kries type of gain adjustment, followed by a power function that has a variable exponent. The exponent of the power function is determined by the overall luminance of the adapting field. In addition to the power function, the Nayatani model adds a noise term and a coefficient for forcing complete color constancy of nonselective (gray) samples of the same luminance as the adapting field. The power function enables the Nayatani model of chromatic adaptation to predict luminance appearance phenomena, such as the Hunt and Stevens effect. The noise term aids in the prediction of threshold data. Equations 2.20 through 2.22 show the generalized expressions of this nonlinear model.

$$L_{adapted} = a_L \cdot \left(\frac{L + L_n}{L_{white} + L_n} \right)^{\beta_L} \tag{2.20}$$

$$M_{adapted} = a_M \cdot \left(\frac{M + M_n}{M_{white} + M_n} \right)^{\beta_M} \tag{2.21}$$

$$S_{adapted} = a_S \cdot \left(\frac{S + S_n}{S_{white} + S_n} \right)^{\beta_S} \tag{2.22}$$

where $L_{adapted}$, $M_{adapted}$, $S_{adapted}$ = adapted cone response signals
L, M, S = input cone response signals
L_{white}, M_{white}, S_{white} = cone responses of the adapting condition
L_n, M_n, S_n = additive noise terms
β_L, β_M, β_S = exponent terms for the power function and are based on the adapting luminance level

In addition to these terms, a_L, a_M, and a_S are coefficients determined to produce color constancy for medium gray stimuli.

The Nayatani model illustrates that a simple extension of a von Kries type of chromatic adaptation model was capable of predicting many complicated color appearance phenomena. This model has served as the basis for many of the other chromatic adaptation and color appearance models that were to follow. More information on this chromatic adaptation model and several of its enhancements can be found in publications by Nayatani et al.[49] and Fairchild.[1]

2.6.7 *Fairchild model*

The original nonlinear Nayatani model suffered slightly from overpredicting the degree of adaptation. That is to say, it predicted more complete adaptation than was witnessed experimentally. Despite many claims that the human visual system is "color constant," often there are situations where

chromatic adaptation is less than 100% complete.[50] This prompted a series of experiments attempting to measure the degree of adaptation for many different forms of adapting stimuli, including both hard and soft copy.[51] These experiments helped derive a linear chromatic adaptation model that accounted for luminance effects, discounting the illuminant, and incomplete adaptation.[52,53]

This model, like the von Kries and Nayatani models before it, is based on a relatively simple extension of basic CIE colorimetry. The general form of this model is similar to the von Kries model, as shown in Equation 2.23.

$$\begin{bmatrix} L_{adapted} \\ M_{adapted} \\ S_{adapted} \end{bmatrix} = \begin{bmatrix} a_L & 0.0 & 0.0 \\ 0.0 & a_M & 0.0 \\ 0.0 & 0.0 & a_S \end{bmatrix} \cdot \begin{bmatrix} L \\ M \\ S \end{bmatrix} \tag{2.23}$$

where a_L, a_M, a_S are adapting gain control coefficients.

These gain control coefficients are calculated in a slightly more complex manner than the typical von Kries method. Equations 2.24 through 2.26 illustrate the calculations for the L cone coefficients. The M and S cone coefficients are calculated in a similar form.

$$a_L = \frac{P_L}{L_n} \tag{2.24}$$

$$P_L = \frac{(1 + Y_n^{1/3} + l_E)}{(1 + Y_n^{1/3} + 1/l_E)} \tag{2.25}$$

$$l_E = \frac{3(L_n/L_E)}{L_n/L_E + M_n/S_n/S_E} \tag{2.26}$$

While daunting at first, these equations really are essentially a modified von Kries transformation. The Y_n term refers to the adapting luminance in cd/m^2. Any term with an n subscript refers to the adapting stimulus, while terms with an E subscript refer to the equal-energy illuminant. Equation 2.24 simplifies to a complete von Kries adaptation term as p approaches 1. On the other hand, Equation 2.24 can also simplify to zero adaptation as p approaches the adapting cone response value. Any value in between represents a degree of incomplete adaptation. The amount of adaptation is a function of both overall luminance level as well as deviation from the equal-energy illuminant. Essentially, as the luminance level increases, so too does the degree of adaptation, and the farther the adapting illuminant is from the equal-energy illuminant, the less adaptation.

The original Fairchild model also included a luminance-dependent interaction among the three cone types. This was subsequently removed, when it was determined to produce an overall increase in lightness predictions.[54] Corresponding color data can be calculated using this model by cascading Equation 2.23 with the Hunt–Pointer–Estevez primaries. The cascaded equation reduces to a simple 3×3 matrix multiplication, allowing for quick calculations for large datasets. For this reason, the Fairchild chromatic adaptation transform, and the color appearance model that was based on it, are useful for processing image data.

2.6.8 *Spectrally sharpened chromatic adaptation models*

Much chromatic adaptation research focus of late has been on the topic of "spectrally sharpened" cone fundamentals.[55–58] The research has been a convergence of two rather distinct fields: color science and computational color constancy. The first chromatic adaptation transform to use spectral sharpened cone fundamentals was the Bradford transform.[55] The Bradford transform is also a modified von Kries gain control model, with a nonlinear term similar to Nayatani's model on the short wavelength cone signal. The calculations in this model begin with a transform from CIE XYZ tristimulus values into normalized cone responses. These calculations are shown in Equations 2.27 and 2.28.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \mathbf{M} \cdot \begin{bmatrix} X/Y \\ Y/Y \\ Z/Y \end{bmatrix} \tag{2.27}$$

$$\mathbf{M} = \begin{bmatrix} 0.8951 & 0.2664 & -0.16614 \\ -0.7502 & 1.7135 & 0.0367 \\ 0.0389 & -0.0685 & 1.0296 \end{bmatrix} \tag{2.28}$$

There are several interesting features of this transform. The XYZ tristimulus values are all normalized by dividing by the Y. This is in effect luminance normalization, as all stimuli with identical chromaticity coordinates will have identical "cone" responses. The cone responses, RGB, do not represent physiologically plausible cone responses. Instead, they represent spectrally sharpened cone responses. What that means is that "cones" themselves have narrower support as well as negative responsivity at some wavelengths. Figure 2.14 illustrates the principle of sharpened sensors. The sharpened responsivities tend to preserve saturation as well as color constancy. The Bradford responsivities are not the only spectrally sharpened cones that can be used in a chromatic adaptation transform. More details can be found in publications by Finlayson[57] and Calabria.[58]

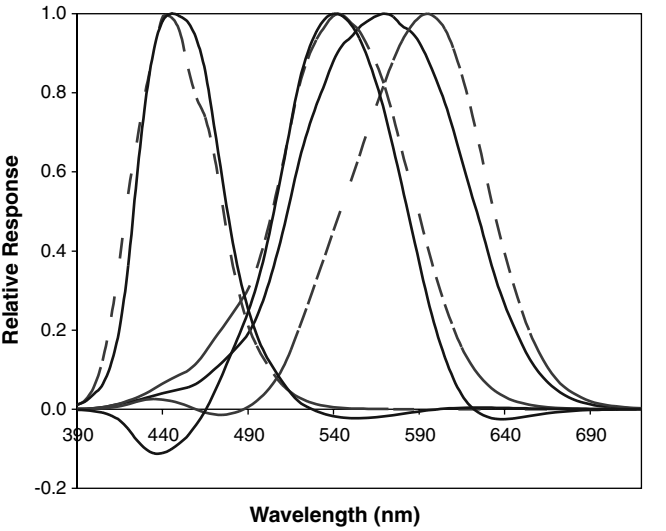


Figure 2.14 A comparison of spectrally sharpened cone responses (solid lines) and physiological cone responses (dashed lines).

The remainder of the Bradford transform is relatively straightforward. With the addition of terms for incomplete adaptation, this is the chromatic adaptation model used in the CIECAM97s color appearance model, so further details are given below.

2.7 Color appearance models

CIE tristimulus colorimetry was designed with a single purpose, for which it has enjoyed good success. This purpose is to predict when two simple stimuli will match, for the average observer, under a single viewing condition. We have already seen the limitations of basic colorimetry with some of the color appearance phenomena described above. Chromatic adaptation transforms, as described in the previous section, extend basic colorimetry so that it is possible to predict matches across disparate viewing conditions. Chromatic adaptation transforms are still limited, in that they do not help describe the actual color appearance of a stimulus.

To accurately describe the color appearance of a stimulus, we must use the color terminology described in an earlier section. These terms include the relative terms of lightness, hue, saturation, and chroma as well as the absolute terms of brightness, colorfulness, and hue (again). Even with a chromatic adaptation transform, CIE tristimulus colorimetry is not able to describe any of these appearance terms. To do that, it is necessary to use a color appearance model.

So what is a color appearance model, exactly? The CIE Technical Committee TC1-34, *Testing Colour Appearance Models*, came up with a definition

of what constitutes a color appearance model.[59] The definition agreed upon is as follows: "A color appearance model is any model that includes predictors of at least the relative color-appearance attributes of lightness, chroma, and hue." This is a relatively lenient definition of what constitutes a color appearance model, though it does require some form of a chromatic adaptation transform at the very least. More complicated models are capable of predicting absolute attributes, such as brightness and colorfulness, as well as luminance-dependent effects, such as the Hunt and Stevens effects. Spatially structured phenomena, such as crispening and simultaneous contrast, require both models of spatial vision as well as color appearance.

Many color appearance models are available, each designed with specific goals in mind. Among those models are CIELAB, Hunt, Nayatani, ATD, RLAB, LLAB, ZLAB, and CIECAM97s. This section will describe CIELAB as a rudimentary color appearance model, as well as CIECAM97s, which is the CIE recommended model. Fairchild[1] has presented a very thorough review of all of these models.

2.7.1 CIELAB as a color appearance model

Although designed as a uniform color space for expressing color differences, rather than a color appearance model, CIELAB does have predictors of lightness, chroma, and hue. These predictors allow CIELAB to be labeled as a color appearance model. We will use it here as a simple model to illustrate the design of more complicated color appearance models.

CIELAB calculations require a pair of CIE XYZ tristimulus values, those of the stimulus itself, as well as those of the reference white point. The reference white point values are used in a von Kries type of chromatic adaptation transform. The adaptation transform is followed by a compressive cube-root nonlinearity and an opponent-color transformation. The exact calculations are shown in Equations 2.29 through 2.32.

$$L^* = 116 f(Y/Y_n) - 16 \tag{2.29}$$

$$a^* = 500[f(X/X_n) - f(Y/Y_n)] \tag{2.30}$$

$$b^* = 200[f(Y/Y_n) - f(Z/Z_n)] \tag{2.31}$$

$$f(x) = \begin{cases} (x)^{1/3} & \text{if} > 0.008856 \\ 7.787(x) + 16/116 & \text{if} \le 0.008856 \end{cases} \tag{2.32}$$

X, Y, and Z are the tristimulus values of the stimulus, while X_n, Y_n, and Z_n are the tristimulus values of the adapting white. Several points need

to be emphasized. The white-point normalization, or chromatic adaptation, is not performed in a physiological cone space. Rather, it is performed in XYZ tristimulus space. This transform is sometimes referred to as a "wrong von Kries" chromatic adaptation transform.[60] The effects of performing the chromatic adaptation in XYZ tristimulus space, rather than cone space, are most noticeable in the hue predictions, often causing inaccurate hue shifts.

The cube root power functions attempt to model the compressive relationship between physical measurements and psychological perceptions. These compressive results were first discussed above in regard to Fechner's and Stevens' laws. The cube root function is replaced by a linear function for very dark stimuli as shown in Equation 2.32.

The CIELAB L^* coordinate, as expressed in Equation 2.29, is a correlate to perceived lightness. It can range between 0.0, for absolute black stimuli, and 100.0, for diffuse white stimuli. The a^* and b^* coordinates approximate, respectively, the red–green and yellow–blue of an opponent color space. A positive a^* value approximates red, while a negative value approximates green. Similarly, a positive b^* correlates to yellow, while negative values correlate to blue. Achromatic stimuli, such as whites, grays, and blacks, have values of 0.0 for both a^* and b^*.

The definition of a color appearance model requires a minimum of predictions for lightness, chroma, and hue. CIELAB L^* provides a lightness prediction, but a^* and b^* do not fully predict correlates of chroma and hue. These correlates can be calculated by transforming the Cartesian coordinates of a^* and b^* into cylindrical coordinates of C^*_{ab} and h_{ab}, where C^*_{ab} represents chroma, and h_{ab} represents hue angle. Equations 2.33 and 2.34 illustrate those transformations.

$$C^*_{ab} = \sqrt{(a^{*2} + b^{*2})} \tag{2.33}$$

$$h_{ab} = \tan^{-1}(b^*/a^*) \tag{2.34}$$

With the cylindrical coordinates of chroma and hue angle, we now have enough information to predict the color appearance of a stimulus, with several caveats, however. The wrong von Kries transform is clearly a source of color appearance errors. CIELAB is also incapable of predicting many of the color appearance phenomena described above. These include all luminance, surround, background, and discounting-the-illuminant effects. CIELAB also assumes 100% adaptation to the white point. Since CIELAB was designed to predict only small color differences between similar objects under a single viewing condition, it is impressive that it can be used as a color appearance model at all. The CIELAB space is also known to have hue nonuniformities, especially in the blue region.[61–63] This becomes important in certain image processing techniques, such as gamut mapping, where

it is desirable to follow lines of constant perceptual hue. Clearly, it is important to have a color appearance model that was designed specifically for the use.

2.7.2 *The genesis of color appearance models*

Color appearance research over the course of many years has resulted in the formulation of many different color appearance models, each with different goals and methods. Until recently, it was often difficult to decide which model to use for any given task. This changed in 1997, with formulation of the CIE-recommended CIECAM97s color appearance model.[59] The CIECAM97s model was designed to work as least as well as, if not better than, all of the previous models for the color appearance phenomena it predicts. Thus, it is essentially a hybrid of the best parts of many different models. It is important to understand the pedigree of CIECAM97s so as to understand why it takes the form it does. This pedigree stems from the Hunt, Nayatani, RLAB, and LLAB models.[59] The interested reader is encouraged to delve into this rich history of color appearance research. The texts by Hunt[36] and Fairchild[1] provide many references to the development of these models.

The *Hunt* model is very sophisticated and designed to predict many color appearance phenomena. It has undergone relentless development over the course of more than two decades.[36] The high degree of sophistication in the model comes at the price of a high degree of complexity. Perhaps the model is better described as a model of the human visual system response. This model was designed to predict a wide range of appearance phenomena, including changes in background, surround, luminance level, and viewing modes. Many of the features, including the underlying color space, found in CIECAM97s are direct descendents of the Hunt model of color appearance.

The *Nayatani* model of color appearance is another model capable of predicting a wide range of appearance phenomena. The Nayatani model evolved directly from the nonlinear chromatic adaptation transform discussed in the previous section. This model has also undergone many revisions over the years. The most recent revisions, as well as a thorough summary, were described by Nayatani et al. in 1995.[64] This model was originally designed as a model for predicting the appearance of objects under various illuminants from an illumination engineering perspective. The ultimate goal of predicting the color rendering properties of light sources is quite different from some of the other color appearance models; therefore, the Nayatani model predicts some phenomena differently than other models, such as those designed with a goal of accurate color image reproduction.

One model designed with color image reproduction in mind is the *RLAB* color appearance model.[65] The RLAB model was developed as a simple color appearance model designed for practical applications. It is based on the

Fairchild incomplete chromatic adaptation transform and is thus capable of predicting many significant color appearance phenomena. The RLAB model was specifically targeted at cross-media image reproduction, such as a CRT to print system, and was built to extend upon CIE colorimetry. Because of its simplicity in design, it is incapable of predicting certain appearance correlates such as brightness and lightness. It is also not designed for use across wide luminance levels and does not predict luminance effects such as the Hunt and Stevens effects.

Another similarly designed model is the *LLAB* color appearance model.[65] This model was designed as a model of color appearance specification, color difference calculation, and color match prediction. Like RLAB, it is designed to extend CIE colorimetry and CIELAB. Built on the Bradford chromatic adaptation, the LLAB model calculates predictions of lightness, chroma, colorfulness, saturation, and hue. It is capable of predicting many appearance phenomena, such as surround and background changes, discounting the illuminant, and the Hunt effect. This model also has a specified color difference equation. It is incapable of predicting the Stevens effect or incomplete chromatic adaptation. The LLAB model is relatively simple, lying between RLAB and the Hunt model in complexity.

2.7.3 CIECAM97s

The existence of many different color appearance models, each derived with different goals and techniques, has led to confusion in both industry and research. Traditionally, it has been difficult to choose which model to use for any given situation, and thus industry acceptance of color appearance models was tenuous at best. The CIE recognized this problem and created TC1-34 with the task of creating a single color appearance model. The goal was to create uniformity of practice with compatibility with modern color imaging systems in mind.[59] TC1-34 was successful in their task with the formulation of the CIE 1997 Interim Color Appearance Model (simple version), CIECAM97s. CIECAM97s is the amalgamation of the research efforts of many people over many years.[59]

The CIECAM97s model requires certain input data. These data include the luminance of the adapting field, expressed in cd/m^2. This is normally taken to be 20% of the luminance of white in the adapting filed and is designated L_a. The CIE tristimulus values of the stimulus in the source conditions, designated X, Y, and Z, as well as the source itself, X_w, Y_w, and Z_w, are also necessary. Additional inputs include the relative luminance of the source background, designated, Y_b, also in the source conditions.

In addition to the above model inputs, several constants need to be selected. These constants include the impact of surround, c; a chromatic induction factor, N_c; a lightness contrast factor, F_{LL}; and a factor determining the degree of chromatic adaptation, F. These constants can be selected using the following chart:

Viewing Condition	c	N_c	F_{LL}	F
Average surround, samples subtending > 4°	0.69	1.0	0.0	1.0
Average surround	0.69	1.0	1.0	1.0
Dim surround	0.59	1.1	1.0	0.9
Dark surround	0.525	0.8	1.0	0.9
Cut-sheet transparencies (on a viewing box)	0.41	0.8	1.0	0.9

2.7.3.1 *Chromatic adaptation*

The first step is to transform the stimulus from the source viewing conditions into the conditions of the equal-energy illuminant. This chromatic adaptation transform uses the spectrally sharpened cone responses, RGB, of the Bradford transform given above. Equations 2.35 and 2.36 illustrate the transformation into the cone primaries as well as the inverse transform.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \mathbf{M} \cdot \begin{bmatrix} X/Y \\ Y/Y \\ Z/Y \end{bmatrix} \tag{2.35}$$

$$\mathbf{M}_B = \begin{bmatrix} 0.8951 & 0.2664 & -0.1614 \\ -0.7502 & 1.7135 & 0.0367 \\ 0.0389 & -0.0685 & 1.0296 \end{bmatrix} \quad \mathbf{M}^{-1}{}_B = \begin{bmatrix} 0.9870 & -0.1471 & 0.1600 \\ 0.4323 & 0.5184 & 0.0493 \\ -0.0085 & 0.0400 & 0.9685 \end{bmatrix} \tag{2.36}$$

The chromatic adaptation transform itself is a modified von Kries type of transformation, with a nonlinear power function on the short wavelength. The degree of adaptation is determined using the variable D. The degree of adaptation is set to 1.0 for complete adaptation or complete discounting the illuminant, as is generally the case for reflecting materials. D is set to 0.0 for no adaptation and can take on intermediate values for various other degrees of adaptation. These values can be manually determined against existing data or can be calculated using Equation 2.41. This calculation is based on luminance levels as well as surround equations.

$$R_c = [D)1.0/R_w + 1 - D]R \tag{2.37}$$

$$G_c = [D(1.0/G_w) + 1 - D]G \tag{2.38}$$

$$B_c = [D(1.0/B_w^p) + 1 - D]|B|^p \tag{2.39}$$

$$p = (B_w/1.0)^{0.0834} \tag{2.40}$$

$$D = F - F/[1 + 2(L_A^{1/4}) + (L_A^2)/300] \tag{2.41}$$

If B happens to be negative, then B_c also must be set to be negative. The above calculations must also be performed for the source white, as they are required in later calculations. Before further calculations can be performed, various other factors must be determined. These factors include the background induction factor, n, the brightness and chromatic induction factors, N_{bb} and N_{cb}, and the base exponential linearity, z. These factors are calculated using Equations 2.42 through 2.46.

$$k = 1/(5L_A + 1) \tag{2.42}$$

$$F_L = 0.2k^4(5L_A) + 0.1(1 - k^4)(5L_A)^{1/3} \tag{2.43}$$

$$n = Y_b/Y_w \tag{2.44}$$

$$N_{bb} = N_{cb} = 0.725(1/n)^{02} \tag{2.45}$$

$$z = 1 + F_{LL}n^{1/2} \tag{2.46}$$

The post-adapted signals for both the sample and the source white point must then be transformed back from the sharpened cone responses into physiological cone responses. This is accomplished using the inverse of the Bradford transform, $\mathbf{M}_B^{-1}$, given in Equation 2.36, and the Hunt–Pointer–Estevez transformation. The complete transformation is given in Equations 2.47 and 2.48.

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \mathbf{M}_H M_B^{-1} \begin{bmatrix} R_c Y \\ G_c Y \\ B_c Y \end{bmatrix} \tag{2.47}$$

$$\mathbf{M}_H = \begin{bmatrix} 0.38971 & 0.68898 & -0.07868 \\ -0.22981 & 1.18340 & 0.04641 \\ 0.00 & 0.00 & 1.00 \end{bmatrix} \quad \mathbf{M}_H^{-1} = \begin{bmatrix} 1.9102 & -1.1121 & 0.2019 \\ 0.3710 & 0.6291 & 0.00 \\ 0.00 & 0.00 & 1.00 \end{bmatrix} \tag{2.48}$$

The signals are then processed through a nonlinear response compression to get post-adaptation cone responses. This is done for both the stimulus and the adapting white. Equations 2.49 through 2.51 illustrate this calculation.

$$R'_a = \frac{40(F_L R'/100)^{0.73}}{[(F_L R'/100)^{0.73} + 2]} + 1 \tag{2.49}$$

$$G'_a = \frac{40(F_L G'/100)^{0.73}}{[(F_L G'/100)^{0.73} + 2]} + 1 \tag{2.50}$$

$$B'_a = \frac{40(F_L B'/100)^{0.73}}{[(F_L B'/100)^{0.73} + 2]} + 1 \tag{2.51}$$

2.7.3.2 *Appearance correlates*

The adapted cone responses are then used to determine correlates of appearance. The first step is to calculate preliminary red–green and yellow–blue opponent dimensions. This is accomplished using Equations 2.52 and 2.53.

$$a = R'_a - 12G'_a/11 + B'_a/11 \tag{2.52}$$

$$b = (1/9)(R'_a + G'_a - 2B'_a) \tag{2.53}$$

The hue angle, h, is calculated in a similar manner as the CIELAB hue angle. This calculation is done using Equation 2.54.

$$h = \tan^{-1}(b/a) \tag{2.54}$$

Often, it is desirable to have the hue correlates for the four unique hues (red, green, yellow, and blue) lie opposite each other in a color space. This is known as the *hue quadrature.* Each of the unique hues has different weights in regard to the perceptual colorization of neutral colors, and this is known as the *hue's eccentricity factor.* Hue quadrature, H, and eccentricity factors, e, are calculated from the following unique hue data via linear interpolation between the following values for the unique hues:

Red	$h = 20.14$	$e = 0.8$	$H = 0$ or 400
Yellow	$h = 90.00$	$e = 0.7$	$H = 100$
Green	$h = 164.25$	$e = 1.0$	$H = 200$
Blue	$h = 237.53$	$e = 1.2$	$H = 300$

An example of the linear interpolation used to calculate hue quadrature and eccentricity values for any given hue angle is shown using Equations 2.55 and 2.56.

$$e = e_1 + (e_2 - e_1)(h - h_1)/(h_2 - h_1) \tag{2.55}$$

$$H = H_1 + \frac{100(h - h_1)/e_1}{(h - h_1)/\mathrm{e}_1 + (h_2 - h)/e_2} \tag{2.56}$$

The achromatic response is then calculated using Equation 2.57. This response is used to calculate brightness and lightness, so the calculations must be performed for both the stimulus and the adapting white.

$$A = [2R'_a + G'_a + (1/20)(B'_a - 2.05)]N_{bb} \tag{2.57}$$

Lightness, J, is then calculated from the achromatic response to both the stimulus and the adapting white. This is shown in Equation 2.58.

$$J = 100(A/A_w)^{cz} \tag{2.58}$$

Brightness is calculated using the lightness value and the achromatic response for the adapting white. Brightness is designated Q and is calculated using Equation 2.59.

$$Q = (1.24/c)(J/100)^{0.67}(A_w + 3)^{09} \tag{2.59}$$

Thus, we now have correlates of hue, brightness, and lightness. From these values, we can calculate correlates of saturation, chroma, and colorfulness, designated s, C, and M, respectively. Equations 2.60 through 2.62 illustrate these calculations.

$$s = \frac{50(a^2 + b^2)^{1/2}100e(10/13)N_cN_{cb}}{R'_a + G'_a + (21/20)B'_a} \tag{2.60}$$

$$C = 2.44s^{0.69}(J/100)^{0.67n}(1.64 - 0.29^n) \tag{2.61}$$

$$M = CF_L^{0.15} \tag{2.62}$$

2.7.3.3 Using the model

As can be seen, the CIECAM97s model is rather complicated. In addition to the complexity of the equations themselves, several constants need to be determined prior to utilizing this model. All these choices can be quite daunting to the casual user. To help alleviate these situations, Moroney has provided usage guideline for CIECAM97s.[67]

Often, it is necessary to invert a color appearance model to predict how a stimulus in one viewing condition might appear when viewed in a different situation. To calculate these colors, it is necessary to invert the model. The nonlinearities in the chromatic adaptation transform mean that CIECAM97s can only be inverted using analytical models. Details on this inversion process can be found in the CIE publication[59] or in the text by Fairchild.[1]

2.7.4 Future directions

The CIE-designated name for CIECAM97s has great significance, as it is called the *Interim* color appearance model. While it is considered the best of what was available at the time of formulation, this does not mean that further development has ceased altogether. Already, the CIE has formulated another technical committee, TC8-01, that is charged with considering potential revisions to CIECAM97s. Fairchild has published a list of proposed changes as well as their implementation details.[68] These changes are designed to simplify the model, fix errors, and add accuracy. Among the proposed changes are:

1. Linearize the chromatic adaptation transform, to facilitate inversion.
2. Fix surround compensation errors.
3. Fix lightness of perfect black.
4. Fix chroma scale expansion for low chroma colors.
5. Add continuously variable surround compensation.

It is expected that a new color appearance model incorporating these and other changes will be approved for testing by the CIE in 2002.

Research on color appearance models will not end with the work of TC8-01, either. Already, research is being conducted on the next generation of color appearance models. Models of spatial and color vision are already appearing, such as the spatial extensions of CIELAB proposed by Zhang and Wandell[69] as well as Johnson and Fairchild.[70] Spatial models of vision and color appearance, such as the multiscale model proposed by Pattanaik et al.[71] will allow for the prediction of the spatially structured appearance phenomena described above. Spatial information is thought to be crucial when dealing with the appearance of digital color images. The future is indeed bright for color appearance research.

References

1. Fairchild, M. D., *Color Appearance Models*, Addison Wesley, Reading MA, 1999.
2. CIE, *International Lighting Vocabulary,* CIE Publ. No. 17.4, Vienna, 1987.
3. Hunt, R. W. G., The specification of color appearance. I. Concepts and terms, *Color Res. Appl.,* 2, 55–68, 1977.
4. Hunt, R. W. G., Colour terminology, *Color Res. Appl.,* 3, 79–87, l978.
5. ASTM, *Standard Terminology of Appearance,* E284-95a, 1995.
6. Hering, E., *Outlines of a Theory of the Light Sense*, Harvard Univ. Press, Cambridge, MA, 1920 (trans. by L.M. Hurvich and D. Jameson, 1964).
7. Kaiser, P. K. and Boynton, R. M., *Human Color Vision,* 2nd ed., Optical Society of America, Washington, D.C., 1996.
8. Wandell, B., *Foundations of Vision*, Sinauer, Sunderland, MA, 1995.
9. Hurvich, L. M., *Color Vision,* Sinauer, Sunderland, MA, 1981.
10. Nayatani, Y., Mori, T., Hashimoto, K., and Sobagaki, H., Comparison of color-appearance models, *Color Res. Appl.,* 15, 1990.

11. Bartleson, C. J. and Grum, F., *Optical Radiation Measurements,* Vol. 5: *Visual Measurements,* Academic Press, Orlando, FL, 1984.
12. Gescheider, G. A., *Psychophysics: Method, Theory, and Application,* 2nd ed., Lawrence Erlbaum Associates, Hillsdale, NJ, 1985.
13. Torgeson, W. S., *Theory and Method of Scaling,* John Wiley & Sons, New York, 958.
14. Thurstone, L. L., *The Measurement of Values,* University of Chicago Press, Chicago, IL, 1959.
15. Engeldrum, P. G., *Psychometric Scaling,* Imcotek Press, Reading, MA, 2000.
16. Fechner, G., *Elements of Psychophysics,* Vol. 1 (trans. by H.E. Adler), Rinehart and Winston, New York, 1966.
17. Stevens, S., To honor Fechner, and repeal his law, *Science,* 133, 80–86, 1961.
18. Braun, K. M. and Fairchild, M. D., Testing five color appearance models for changes in viewing conditions, *Color Res. Appl.* 21, 165-173, 1997.
19. OSA, *The Science of Color,* Optical Society of America, Washington, D.C., 145–171, 1963.
20. Albers, J., *Interaction of Color,* Yale University Press, New Haven, CT, 1963.
21. Robertson, A., ISCC annual meeting, 1996.
22. Shevell, S. and Wei, Chromatic induction with remote chromatic contrast varied in magnitude, spatial frequency, and chromaticity, *Vision Res.,* 38, 1561–1566, 1998.
23. Kuehni, R. G., *Color: An Introduction to Practice and Principles,* John Wiley & Sons, New York, 1997.
24. Bressan, P., Neon color spreading: a review, *Perception*, 26, 1353–1366, 1997.
25. Pinna, B., Brelstaff, G., and Spillmann, L., Surface color from boundaries: a new "watercolor" illusion, *Vision Res.,* 41, 2669–2676, 2001.
26. Semmelroth, C. C., Prediction of lightness and brightness on different backgrounds, *J. Opt. Soc. Am.,* 60, 1685–1689, 1970.
27. Moroney, N., Chroma scaling and crispening, in *Proc. IS&T/SID Ninth Color Imaging Conference*, 2001, 97–101.
28. Hunt, R. W. G., Light and dark adaptation and the perception of color, *J. Opt. Soc. Am.,* 42, 190–199, 1952.
29. Stevens, J. C. and Stevens, S. S., Brightness functions: effects of adaptation, *J. Opt. Soc. Am.,* 53, 375–385, 1963.
30. Fairchild, M. D. and Pirrotta, E., Predicting the lightness of chromatic object colors using CIELAB, *Color Res. Appl.,* 16, 385–393, 1991.
31. Purdy, D. M., Spectral hue as a function of intensity, *Am. J. Psychol.*, 43, 541–559, 1931.
32. Hunt, R. W. G., Hue shifts in unrelated and related colors, *Color Res. Appl.,* 14, 235–239, 1989.
33. Helson, H., Fundamental problems in color vision. I. The principle governing changes in hue, saturation, and lightness of non-selective samples in chromatic illumination, *J. Exp. Psychol.* 23, 439–477, 1938.
34. Mori, L., Sobagaki, H., Komatsubara, H., and Ikeda, K., Field trials of CIE chromatic adaptation formula, *Proc. CIE 22nd Session*, Melbourne, Australia, 1991, 55–58.
35. Bartleson, C. J. and Breneman, E. J., Brightness perception in complex fields, *J. Opt. Soc. Am.,* 57, 953–957, 1967.
36. Hunt, R. W. G., *The Reproduction of Colour,* 6th ed., Fountain Press, U.K., 2002.

37. Fairchild, M. D., Testing colour-appearance models: guidelines for coordinated research, *Color Res. Appl.*, 20, 262–267, 1995.
38. Bartleson, C. J., Optimum image tone reproduction, *J. SMPTE*, 84, 613–618, 1975.
39. Jameson, D. and Hurvich, L. M., Essay concerning color constancy, *Ann. Rev. Psychol.*, 40, 1–22, 1989.
40. von Kries, J., *Chromatic Adaptation,* Festschrift der Albrecht-Ludwig-Universitat, Friboug, 1902.
41. MacAdam, D. L., Chromatic adaptation, in *Sources of Color Science,* MIT Press, Cambridge, MA, 1970.
42. Hunt, R. W. G., Pitt, I. T., and Winter, L. M., The preferred reproduction of blue sky, green grass and Caucasian skin in colour photography, *J. Phot. Sci.*, 22, 144–150, 1974.
43. Evans, R. M., *An Introduction to Color*, John Wiley & Sons, New York, 1948.
44. Davidoff, J., *Cognition through Color*, MIT Press, Cambridge, MA, 1991.
45. Hunt, R. W. G. and Pointer, M. R., A colour-appearance transform for the CIE 1931 Standard Colorimetric Observer, *Color Res. Appl.,* 10, 165–179, 1985.
46. Nayatani, Y., Takahama, K., and Sobagaki, H., Formulation of a nonlinear model of chromatic adaptation, *Color Res. Appl.,* 6, 161–171, 1981.
47. Nayatani, Y., Takahama, K., Sobagaki, H., and Hirono, J., On exponents of a nonlinear model of chromatic adaptation, *Color Res. Appl.,* 7, 34–45, 1982.
48. MacAdam, D. L., A nonlinear hypothesis for chromatic adaptation, *Vis. Res.*, 1, 9–41, 1961.
49. Nayatani, Y., Hashimoto, K., Takahama, K., and Sobagaki, H., A nonlinear color-appearance model using Estevez–Hunt–Pointer primaries, *Color Res. Appl.,* 12, 231–242, 1987.
50. Breneman, E. J., Corresponding chromaticities for different states of adaptation to complex visual fields, *J. Opt. Soc. Am.*, A4, 1115–1129, 1987.
51. Fairchild, M. D., Chromatic Adaptation and Color Appearance, Ph.D. dissertation, University of Rochester, NY, 1990.
52. Fairchild, M. D., A model of incomplete chromatic adaptation, *Proc. 22nd Session of the CIE*, Melbourne, 1991, 33–34.
53. Fairchild, M. D., Formulation and testing of an incomplete-chromatic-adaptation model, *Color Res. Appl.,* 16, 243–250, 1991.
54. Fairchild, M. D., Pirrotta, E., and Kim, T. G., Successive-Ganzfeld haploscopic viewing technique for color-appearance research, *Color Res. Appl.,* 19, 214–221, 1994.
55. Lam, K. M., Metamerism and Colour Constancy, Ph.D. thesis, University of Bradford, 1985.
56. Finlayson, G. D., Drew, M. S., and Funt, B. V., Spectral sharpening: Sensor transformations for improved color constancy, *J. Opt. Soc. Am.,* A11, 1553–1563, 1994.
57. Finlayson, G. D. and Süsstrunk, S., Performance of a chromatic adaptation transform based on spectral sharpening, *Proc. IS&T/SID 8th Color Imaging Conf.*, 2000, 56–60.
58. Calabria, A. J. and Fairchild, M. D., Herding CATs: A comparison of linear chromatic-adaptation transforms for CIECAM97s, *Proc. IS&T/SID 9th Color Imaging Conf.*, 174–178, 2001.
59. CIE TC1-34 Final Report, *The CIE 1997 Interim Colour Appearance Model (Simple Version),* CIECAM97s, 1998.

60. Terstiege, H., Chromatic adaptation: a state-of-the-art report, *J. Col. Appear.*, 1, 19–23, 1972.
61. Hung, P. and Berns, R. S., Determination of constant hue loci for a CRT gamut and their predictions using color appearance spaces, *Color Res. Appl.*, 20, 285–295, 1995.
62. Ebner, F. and Fairchild, M. D., Finding constant hue surfaces in color space, *Proc. of SPIE, Color Imaging: Device Independent Color, Color Hardcopy and Graphic Arts III,* 3300–16, 1998, 107–117.
63. Braun, G. and Fairchild, M. D., Color gamut mapping in a hue-linearized CIELAB color space, *Proc. IS&T's 6th CIC Conf.*, 1998, 163–168.
64. Nayatani, Y., Sobagaki, H., Hashimoto, K., and Yano, Y., Lightness dependency of chroma scales of a nonlinear color-appearance model and its latest formulation, *Color Res. Appl.,* 20, 156–167, 1995.
65. Fairchild, M. D. and Berns, R. S., Image color appearance specification through extension of CIELAB, *Color Res. Appl.,* 18, 178–190, 1993.
66. Luo, M. R., Lo, M. C., and Kuo, W. G., The LLAB(l:c) colour model, *Color Res. Appl.*, 21, 412–429, 1996.
67. Moroney, N., Usage guidelines for CIECAM97s, in *Proc. IS&T's PICS Conf.*, 2000.
68. Fairchild, M. D., A revision of CIECAM97s for practical applications, *Color Res. Appl.,* 26, 418–427, 2001.
69. Zhang, X. M. and Wandell, B. A., A spatial extension to CIELAB for digital color image reproduction, in *Proc. SID Symposiums*, l996.
70. Johnson, G. M. and Fairchild, M. D., Darwinism of color image difference models, in *Proc. IS&T/SID 9th Color Imaging Conference*, 2001, 108–112.
71. Pattanaik, S. N., Fairchild, M. D., Ferwerda, J.A., and Greenberg, D. P., Multiscale model of adaptation, spatial vision, and color appearance, in *Proc. IS&T/SID 6th Color Imaging Conference*, Scottsdale, AZ, 1998, 2–7.

chapter three

Physical models for color prediction

Patrick Emmel†
Clariant International

Contents

† This work was done while the author was at the Ecole Polytechnique Fédérale de Lausanne (EPFL).

3.1 Introduction

Numerous physical phenomena influence color: the light source, surface reflection, light absorption, light scattering, reflection on the substrate, multiple-internal reflections at the ink–air interface, and the combination of several light-absorbing and light-scattering substances. In the particular case of halftone prints, additional effects, such as the optical dot gain (also called Yule–Nielsen effect), must also be taken into account. This makes accurate color prediction very difficult. Until recently, the physical phenomena involved were described separately by several classical models: Lambert's law for diffuse light sources, the Fresnel reflection law, Beer's absorption law, the Saunderson correction for multiple internal reflections, and the Kubelka–Munk model for absorbing and scattering media. The colors of halftone prints were predicted using other theories: the Murray–Davis model, the Neugebauer model, the Yule–Nielsen model, and the Clapper–Yule model for optical dot gain.

This chapter is based, however, on a new global approach that incorporates all the physical contributing phenomena listed above into a single model using a mathematical framework based on matrices. Classical results (for example, the Murray–Davis equation, the Clapper–Yule relation, or the Kubelka–Munk model) correspond to particular cases of this model. Furthermore, the model we present here predicts accurately the spectra of printed color samples (uniform or halftoned), and it can be used for any inks or colorants, including the standard Cyan, Magenta, Yellow, and Black (CMYK) that are usually used in printing devices or any other nonstandard inks.

Throughout this chapter we will consider the entities $T(\lambda)$, $R(\lambda)$, $D(\lambda)$, ... as spectra (transmission spectrum, reflection spectrum, density spectrum, etc.). But, when the wavelength notation (λ) is omitted, we will consider them as coefficients (transmission coefficient, reflection coefficient, density, etc.). We will consider these terms interchangeably as synonyms according to our best convenience.

We start our presentation with a few useful definitions from radiometry given in Section 3.2. Sections 3.3, 3.4, and 3.5 introduce the basic physical

laws of light reflection, light absorption, and light scattering, respectively. The complex interaction between light and matter forces us to use phenomenological models, which are presented in Section 3.6. The particular case of fluorescent media is discussed in Section 3.7. Traditional models used to predict the color of halftone prints are presented in Section 3.8. Finally, we explain in Section 3.9 how all these models can be incorporated into a single model for the prediction of halftones.

3.2 A few results from radiometry

Let us start with the definition of a few radiometric quantities and terms.[1]

- A surface element of area da receiving a light flux $d\phi_r$ is said to be under an *irradiance* E (unit $W \cdot m^{-2}$).

$$E = \frac{d\phi_r}{da} \tag{3.1}$$

- A surface element of area da emitting a light flux $d\phi_e$ has an *exitance* M (unit $W \cdot m^{-2}$).

$$M = \frac{d\phi_e}{da} \tag{3.2}$$

- A surface element of area da is said to be of *radiance* L (unit $W \cdot sr^{-1} \cdot m^{-2}$) if it emits a flux $d^2\phi_e$ in a solid angle $d\omega$ making an angle θ with the normal to the surface.

$$L = \frac{d^2\phi_e}{\cos\theta d\omega da} \tag{3.3}$$

- The *intensity* I (unit $W \cdot sr^{-1}$) of a light flux $d\phi$ in a solid angle $d\omega$ is defined by

$$I = \frac{d\phi}{d\omega} \tag{3.4}$$

- A light source whose radiance L is constant in all space directions is said to be *Lambertian*. The exitance M of such a source is $M = \pi L$ (see Reference 2).
- A diffuse reflector of constant radiance L is said to be a *Lambert surface*[3] or a *Lambertian reflector*. The exitance M of such a surface equals its irradiance E, so $L = E/\pi$ and we have

$$\frac{d^2\phi_e}{\cos\theta d\omega da} = \frac{E}{\pi} \tag{3.5}$$

This formula is called *Lambert's cosine law*. Note that, in the literature,[4] Lambert's cosine law is often presented in terms of intensity,

$$I(\theta) = \frac{d\phi_e}{d\omega} = \frac{E}{\pi} A \cos\theta = I_0 \cos\theta \tag{3.6}$$

where A is the area of the surface.

Equation 3.6 implies that the *radiation pattern* of such a surface (i.e., the locus of the extremities of the intensity vectors, also called *indicatrix*) is a circle (see Figure 3.1). Another useful result is Lambert's cosine law for a conical light beam. The flux emitted in the direction given by the angle θ can be computed by considering the solid angle $d\omega = 2\pi \sin\theta d\theta$ (see Figure 3.2). By replacing $d\omega$ in Equation 3.6, we deduce that a surface A that receives from the upper hemisphere a total flux $\phi_r = \int_S E ds$ emits a diffuse flux ϕ_e whose angular distribution is given by[5]

$$\frac{1}{\phi_r} \frac{\partial \phi_e}{\partial \theta} = 2 \sin\theta \cos\theta = \sin 2\theta \tag{3.7}$$

Natural light has a rather diffuse behavior in which rays do not have a privileged orientation. Therefore, it is useful to define the term *diffuse irradiation*.

- Let A be an opening in an opaque plane (see Figure 3.3). The opening is said to be under a *diffuse irradiation* from the upper hemisphere if A is a Lambertian source in the lower hemisphere. From the point

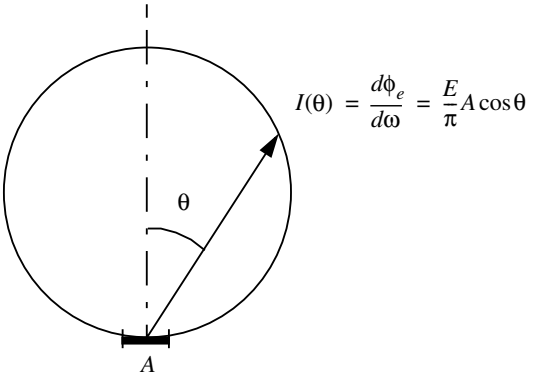


Figure 3.1 According to Lambert's cosine law, the intensity I of the light emitted by a diffuse reflector of area A depends only on the cosine of the angle θ of observation (note that the radiation pattern of such a reflector is a circle).

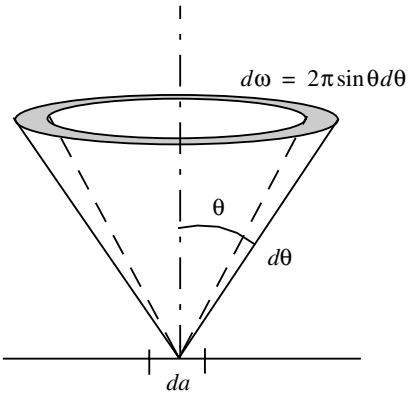


Figure 3.2 Solid angle that must be used when considering a conical light beam emitted by a Lambertian source or reflector.

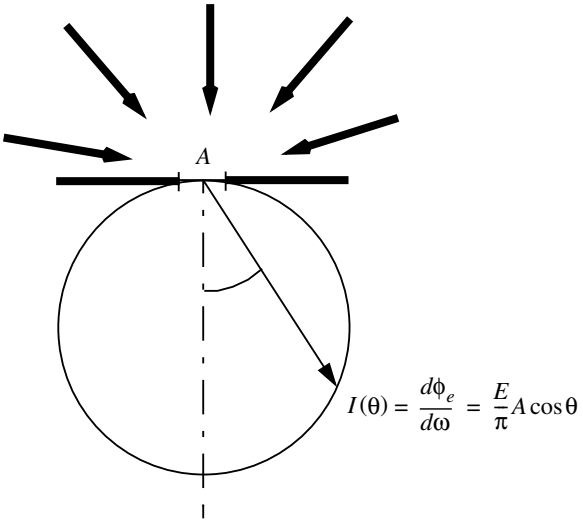


Figure 3.3 Surface element A under diffuse illumination. If A is an opening in an opaque plane that separates the upper and the lower hemispheres, the surface element A is a Lambertian source in the lower hemisphere.

of view of the upper hemisphere, the surface A is said to be a *Lambertian receiver*; its reception pattern or indicatrix is a circle.

Finally, light sources are classified by the way they produce light. *Incandescent* sources produce light by thermal blackbody or near-blackbody radiation. All nonthermal light production is called *luminescence*. Hence, a *luminescent medium* is a medium that produces light by any means except thermal excitation. A common kind of luminescence is photoluminescence, which

includes fluorescence and phosphorescence. A photoluminescent medium produces light when excited with photons. An extended list of forms of luminescence is given in Reference 6.

3.3 Reflection and refraction

The term *reflection* refers to all interaction processes of light with matter in which the photons are sent back into the hemisphere of the incident light. We distinguish between two types of reflection: *specular reflection*, which occurs on smooth surfaces (where the irregularities are small compared to the wavelength of the incident light), and *diffuse reflection*, which occurs on rough surfaces. A beam of light incident on a smooth surface is re-emitted as a well-defined beam, whereas, on a rough surface, it is re-emitted as a multitude of rays emerging in different directions. We define the *reflectance spectrum* (or *reflection coefficient*) $R(\lambda)$ as the ratio of the reflected light flux to the incident light flux for a given wavelength λ.

The term *refraction* refers to the change of direction of a light beam when entering a medium in which the speed of light is different. Therefore, we define the refractive index n of a given medium as the ratio of the speed of light in empty space to the speed of light in the medium. For example, the refractive index is $n = 1.33$ for water, $n = 1.52$ for crown glass, and $n = 2.907$ for rutile (TiO_2).[7] A medium that also attenuates electromagnetic waves has a complex refractive index $\hat{n} = n(1 + i\kappa)$, where κ is the *attenuation index*. For example, the complex refractive index is $\hat{n} = 1.44 + i5.23$ for aluminium, $\hat{n} = 1.51 + i1.63$ for iron, and $\hat{n} = 2.63 + i3.54$ for platinum.[8] Note that, in a metal, light is so intensely attenuated that it can penetrate to a depth of only a few hundred atoms.

3.3.1 Basic laws

Let us recall the basic laws of light refraction and light reflection. A light beam that, with an incidence angle θ_1, hits a refractive surface that separates two media of refractive indices n_1 and n_2 is partially reflected into the first medium and partially refracted into the second medium (see Figure 3.4). The incident beam, the reflected beam, and the refracted beam lie in the same plane, called the *plane of incidence*. The reflected beam makes with the normal to the surface the same angle θ_1 as the incident beam, whereas the

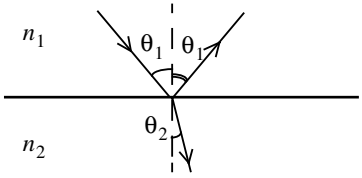


Figure 3.4 Reflection and refraction of a light beam at a refractive surface.

refracted beam makes with the normal an angle θ_2, which is related to θ_1 by *Snell's law.*

$$n_1 \sin(\theta_1) = n_2 \sin(\theta_2) \tag{3.8}$$

Note that, for $n_1 > n_2$, there is a critical angle $\theta_{1max} = \text{asin}(n_2/n_1)$ above which the incident beam is totally reflected.

The intensity of the reflected beam is calculated by considering two polarized electromagnetic waves. One is polarized in parallel to the plane of incidence, and the other is polarized perpendicularly. It can be shown[9] that the reflection coefficient r_a for the parallel polarized wave and the reflection coefficient r_e for the perpendicularly polarized wave are given by

$$r_a = \left(\frac{n_2 \cos\theta_1 - n_1 \cos\theta_2}{n_2 \cos\theta_1 + n_1 \cos\theta_2}\right)^2 \quad \text{and} \quad r_e = \left(\frac{n_1 \cos\theta_1 - n_2 \cos\theta_2}{n_1 \cos\theta_1 + n_2 \cos\theta_2}\right)^2 \tag{3.9}$$

where θ_1 = angle of incidence

θ_2 = angle of refraction according to Snell's law (see Equation 3.8 and Figure 3.4)

In the literature, these relations are known as the *Fresnel relations*. We denote by $r_{n_1, n_2}(\theta_1)$ the reflection coefficient of a beam propagating in a medium of refractive index n_1, which has an incidence angle θ_1 with the refractive surface delimiting a medium of index n_2. Because natural light can be considered as an equal mixture of both types of waves, its reflection coefficient is the mean value of r_a and r_e,

$$r_{n_1, n_2}(\theta_1) = \frac{r_a + r_e}{2} \tag{3.10}$$

Note that Snell's law is also valid for complex refractive indices $\hat{n}$, such as the refractive indices of metals. In the particular case of a light beam having normal incidence on a medium of complex refractive index $\hat{n}$, the reflection coefficient is[10]

$$r_{1, \hat{n}}(0) = \left|\frac{\hat{n} - 1}{\hat{n} + 1}\right|^2 \tag{3.11}$$

For non-normal incidence ($\theta_1 \neq 0$), the angle θ_2 is complex, meaning that a phase change occurs on reflection. For example, a linearly polarized light is reflected as elliptically polarized light. The generalized Fresnel's formulas for complex refractive indices are beyond the scope of this book, but a detailed presentation can be found in the literature.[10]

3.3.2 *Interface reflection under diffuse light*

Let us now compute an average reflection coefficient $\overline{r_{n_1,n_2}}$ for diffuse light arriving on a plane refractive surface. Such an average is calculated by integrating, over all directions, the product of the angular distribution of diffuse irradiation given by Equation 3.7 and the reflection coefficient of a natural light beam given by Equation 3.10.

$$\overline{r_{n_1,n_2}} = \int_0^{\frac{\pi}{2}} (r_{n_1,n_2}(\theta) \cdot \sin 2\theta) d\theta \tag{3.12}$$

This calculation was done by Judd[11] in 1942 for a large number of refractive indices. A typical result is the particular case of the *surface reflection* r_s between air (whose refractive index is $n_1 = 1$) and a medium of refractive index n.

$$r_s = \int_0^{\frac{\pi}{2}} (r_{1,n}(\theta) \cdot \sin 2\theta) d\theta \tag{3.13}$$

In the particular case of a plastic medium, we have $n = 1.5$. The computation of Equation 3.13 leads in this case to $r_s = 0.0918$. This value expresses the average reflection coefficient under a perfect diffuse illumination. When diffuse light crosses the refractive surface in the other direction, from a medium of refractive index n to the air, the reflection occurs within the material medium, and it is therefore called the *internal reflection* r_i.

$$r_i = \int_0^{\frac{\pi}{2}} (r_{n,1}(\theta) \cdot \sin 2\theta) d\theta \tag{3.14}$$

The numerical result of the computation for $n = 1.5$ is, in this case, $r_i = 0.5963$. The numerical results for other refractive indices are given in Table 3.1.

Note that for diffuse light, internal reflection values are always much higher than surface reflection values. When the first medium has a higher refractive index than the second, there is a critical incidence angle, according to Snell's law, above which light is totally reflected. This total reflection is responsible for the high values of $\overline{r_{n_1,n_2}}$ when $n_1 > n_2$.

3.4 *Light absorption*

The term *light absorption* refers to all processes that reduce the intensity of a light beam when interacting with matter. We must distinguish between *true absorption*, where radiative energy is transformed into another kind of energy

Table 3.1 Reflection Coefficient at Normal Incidence $r_{1,n}(0)$, Surface Reflection Coefficient r_s, and Internal Reflection Coefficient r_i for Various Refractive Indices n

n	$r_{1,n}(0)$	r_s	r_i	n	$r_{1,n}(0)$	r_s	r_i
1.0	0.0000	0	0	1.3	0.0170	0.0611	0.4445
1.01	0.0000	0.0031	0.0228	1.31	0.0180	0.0627	0.4538
1.02	0.0001	0.0061	0.0446	1.32	0.0190	0.0643	0.4630
1.03	0.0002	0.0088	0.0657	1.33	0.0201	0.0659	0.4719
1.04	0.0004	0.0114	0.0860	1.34	0.0211	0.0675	0.4807
1.05	0.0006	0.0139	0.1056	1.35	0.0222	0.0691	0.4892
1.06	0.0008	0.0163	0.1245	1.36	0.0233	0.0706	0.4975
1.07	0.0011	0.0186	0.1428	1.37	0.0244	0.0722	0.5057
1.08	0.0015	0.0208	0.1605	1.38	0.0255	0.0737	0.5136
1.09	0.0019	0.0230	0.1777	1.39	0.0266	0.0753	0.5214
1.1	0.0023	0.0252	0.1943	1.4	0.0278	0.0768	0.5290
1.11	0.0027	0.0272	0.2105	1.41	0.0289	0.0783	0.5364
1.12	0.0032	0.0293	0.2261	1.42	0.0301	0.0799	0.5437
1.13	0.0037	0.0313	0.2413	1.43	0.0313	0.0814	0.5508
1.14	0.0043	0.0332	0.2561	1.44	0.0325	0.0829	0.5577
1.15	0.0049	0.0351	0.2704	1.45	0.0337	0.0844	0.5645
1.16	0.0055	0.0370	0.2843	1.46	0.0350	0.0859	0.5711
1.17	0.0061	0.0389	0.2979	1.47	0.0362	0.0873	0.5777
1.18	0.0068	0.0407	0.3110	1.48	0.0375	0.0888	0.5840
1.19	0.0075	0.0425	0.3238	1.49	0.0387	0.0903	0.5902
1.2	0.0083	0.0443	0.3363	1.5	0.0400	0.0918	0.5963
1.21	0.0090	0.0460	0.3484	1.51	0.0413	0.0932	0.6023
1.22	0.0098	0.0478	0.3602	1.52	0.0426	0.0947	0.6082
1.23	0.0106	0.0495	0.3717	1.53	0.0439	0.0962	0.6139
1.24	0.0115	0.0512	0.3829	1.54	0.0452	0.0976	0.6195
1.25	0.0123	0.0529	0.3939	1.55	0.0465	0.0991	0.6250
1.26	0.0132	0.0546	0.4045	1.56	0.0479	0.1005	0.6304
1.27	0.0141	0.0562	0.4149	1.57	0.0492	0.1020	0.6357
1.28	0.0151	0.0579	0.4250	1.58	0.0505	0.1034	0.6408
1.29	0.0160	0.0595	0.4348	1.59	0.0519	0.1048	0.6459

(thermal agitation energy, ionization energy, etc.), and *apparent absorption*, which is due to light scattering (see Section 3.5). To avoid confusion, we will use the word *extinction* for all light intensity reducing processes and reserve the word *absorption* for true absorption.

The absorption mechanisms are different in gases, liquids, and solids. For gases and liquids, there are two main mechanisms. The first mechanism is a quantified change of the energy state of the molecules or the atoms, producing *line spectra*. The second mechanism is the dissociation of

molecules or ionization of atoms, which produces *continuous spectra* with an energy threshold. In solids, the behavior depends on the arrangement of the atoms.[12] The absorption of photons in isolators and ionic crystals induces quantified changes of state that produce *narrow band spectra*. In semiconductors, the electrons require a small amount of energy to jump over the energy gap between the valence and the conduction band. Therefore, photons are absorbed only if their energy is higher than the band gap. Such materials have a continuous absorption spectrum with a threshold. In metals, the electrons belonging to the conduction band move almost freely in the whole volume, acting as a free electron gas. The absorption of photons is so strong over the whole spectrum that light is reflected. The reason is that incident light is an electromagnetic wave that induces an alternating electrical current in the conducting material. According to Maxwell's theory, this current re-emits light out of the metal. As far as alloys are concerned, there is no general rule. Their behavior depends on their crystal structures.

The most widely known classical model for the absorption of light is the *Beer–Lambert–Bouguer law* (which is also called, by abuse of language, *Beer's law*).[13] This model describes the intensity variation of a collimated light beam crossing a medium that contains identical light-absorbing particles at a concentration c. Let us consider an infinitely thin slice of thickness dx of this medium (see Figure 3.5). The model relies on the assumption that the absorbing particles are independent. According to the Beer–Lambert–Bouguer law, the intensity variation $d\phi$ of the light flux that crosses this slice is proportional to the concentration c, to the flux intensity ϕ of the light beam, and to the thickness dx of the slice. Hence, the flux of a collimated light beam that crosses the infinitely thin layer varies as follows:

$$d\phi = -\varepsilon(\lambda) c \phi \ln(10) dx \tag{3.15}$$

where λ = the light wavelength, and the proportionality coefficient
$\varepsilon(\lambda)$ = the *molar decadic absorption coefficient* (or, in short, *absorption coefficient*) of the absorbing particle (unit: $m^2 \cdot mol^{-1}$)

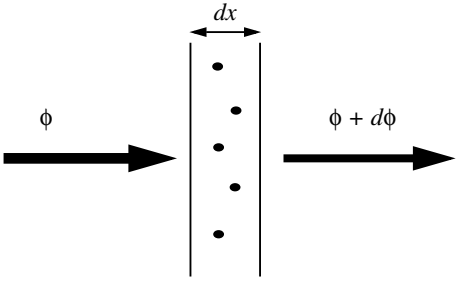


Figure 3.5 Absorption of light by an infinitely thin layer containing light-absorbing particles at a concentration c.

The absorption coefficient $\varepsilon(\lambda)$ can be interpreted as the absorption cross-section area of a mole of particles. For particles of radius r, we have

$$\varepsilon(\lambda) = \frac{N_A}{\ln(10)} \pi r^2 \chi_{abs}(\lambda) \tag{3.16}$$

where $N_A = 6.022 \cdot 10^{23}$ = the Avogadro number
$\chi_{abs}(\lambda)$ = the absorption efficiency factor of the particle

Relation 3.15 is a linear differential equation of the first order whose solution is given by an exponential function. This kind of function will play a central role in our discussion. The integration of Equation 3.15 through a layer of thickness X leads to

$$\phi(X) = \exp[-Xc\varepsilon(\lambda)\ln 10] \cdot \phi(0) = 10^{-[Xc\varepsilon(\lambda)]} \cdot \phi(0) \tag{3.17}$$

The *transmittance spectrum* (or in short *transmittance*) $T(\lambda)$ is then defined as the ratio between the outcoming flux $\phi(X)$ and the incoming flux $\phi(0)$.

$$T(\lambda) = \frac{\phi(X)}{\phi(0)} = \exp[-Xc\varepsilon(\lambda)\ln 10] = 10^{-[Xc\varepsilon(\lambda)]} \tag{3.18}$$

The value $T(\lambda) = 1$ corresponds to a transparent medium, whereas the value $T(\lambda) = 0$ means that no light is transmitted, in which case the medium is said to be opaque. Beer's law is often expressed in a logarithmic form,

$$D(\lambda) = -\log_{10} T(\lambda) = X \cdot c \cdot \varepsilon(\lambda) \tag{3.19}$$

where $D(\lambda)$ is the (optical) *density spectrum* (or *absorption spectrum*), which corresponds to the transmittance $T(\lambda)$. In the density scale, $D(\lambda) = 0$ corresponds to a transparent medium, and the values of $D(\lambda)$ increase logarithmically when the transparency decreases. The extreme case of a totally opaque medium corresponds to an infinite density, $D(\lambda) = \infty$. The transmission spectra and the corresponding density spectra of cyan, magenta, and yellow inks at various concentrations are given in Figures 3.6, 3.7, and Figure 3.8, respectively.

In a mixture of several different absorbing substances that do not interact, the density of the mixture equals the sum of the densities of the individual substances.

$$D(\lambda) = \sum_j D_j(\lambda) \tag{3.20}$$

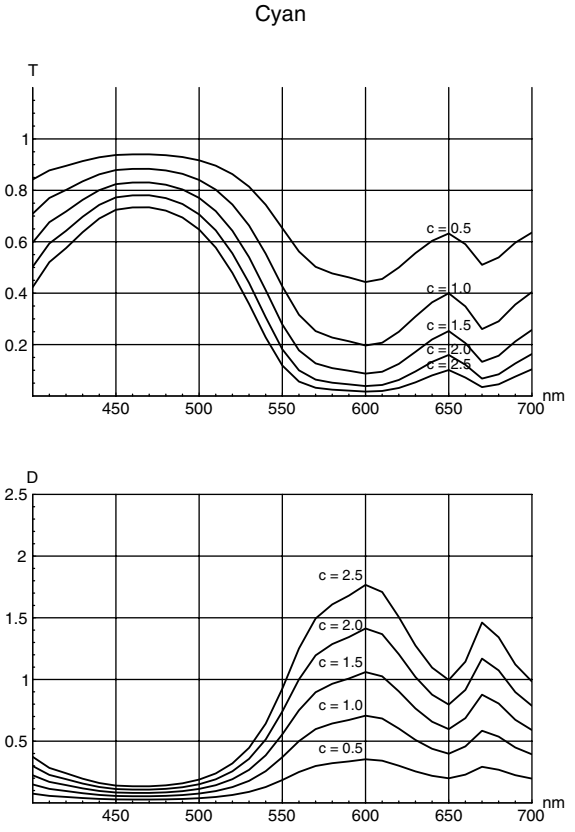


Figure 3.6 Transmission and density spectra of a cyan ink at various concentrations c.

This equation can also be written using the transmittances of the individual substances. The total transmittance equals the product of all transmittances.

$$T(\lambda) = \prod_j T_j(\lambda) \tag{3.21}$$

In practice, Equation 3.21 allows us to compute the transmittance of the mixture of purely light-absorbing inks.

Finally, let us calculate the average density $\overline{D}(\lambda)$ of an infinitely thin slice of an absorbing medium under diffuse illumination (see Figure 3.9). We know the angular distribution of the diffuse light flux from Equation 3.7, and Beer's law gives the absorption in the direction θ, which equals

$$\frac{dx}{\cos\theta} \cdot \varepsilon(\lambda) \cdot c$$

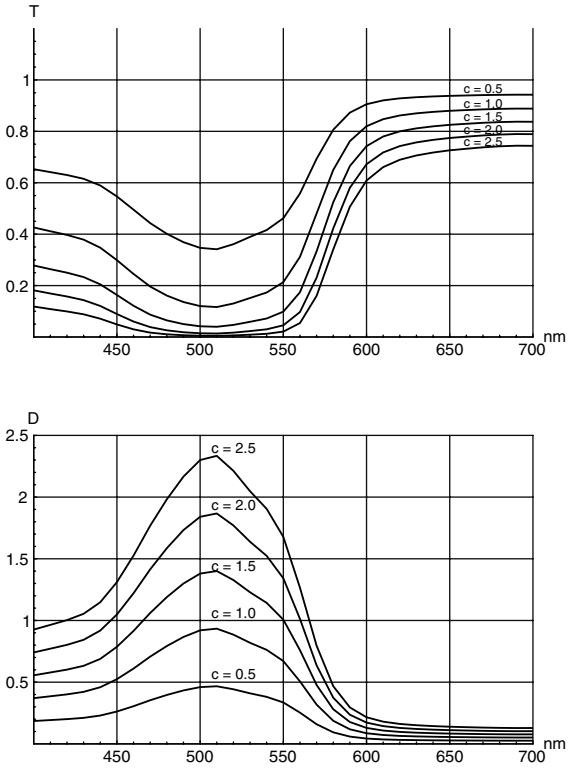


Figure 3.7 Transmission and density spectra of a magenta ink at various concentrations c.

Hence, the average density $\overline{D}(\lambda)$ can be computed by integrating over all directions.

$$\begin{aligned}\overline{D}(\lambda) &= \int_0^{\frac{\pi}{2}} \left(\frac{dx}{\cos\theta} \cdot \varepsilon(\lambda) \cdot c \cdot \sin 2\theta \right) d\theta \\ &= (2 \cdot dx \cdot \varepsilon(\lambda) \cdot c) \int_0^{\frac{\pi}{2}} \sin\theta d\theta \\ &= 2 \cdot dx \cdot \varepsilon(\lambda) \cdot c\end{aligned} \tag{3.22}$$

This shows that the optical density under diffuse illumination is twice the density observed for a collimated light beam. This fundamental result gives us a generalization of Beer's law for diffuse light.

$$d\phi = -2\varepsilon(\lambda) c \phi \ln 10 \cdot dx \tag{3.23}$$

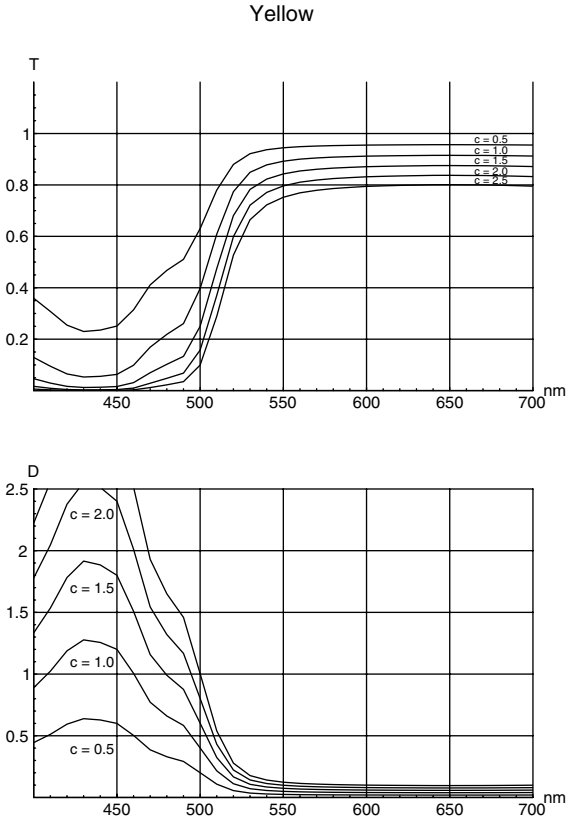


Figure 3.8 Transmission and density spectra of a yellow ink at various concentrations c.

3.5 *Light scattering*

The term *light scattering* refers to all physical processes that move photons apart in different directions. This phenomenon is often caused by local variations of the refractive index within a heterogeneous medium. Other scattering processes, as for instance the Raman effect and the Brillouin scattering, also change the wavelength (i.e., the energy) of the incident photon, but these phenomena are rare in nature.

In this chapter, we will be interested in scattering caused by small particles that are dispersed in a homogeneous medium. Let us consider a thin slice of thickness dx of this scattering medium (see Figure 3.10). The variation $d\phi$ of the collimated light flux that crosses this slice is proportional to the flux intensity ϕ of the light beam and to the thickness dx of the slice.

$$d\phi = -\beta(\lambda)\phi dx \tag{3.24}$$

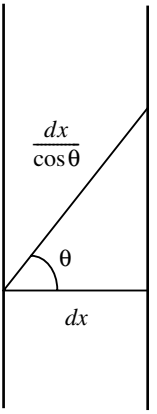


Figure 3.9 The average absorption of an infinitely thin slice under diffuse illumination is related to the average path of the light in the medium.

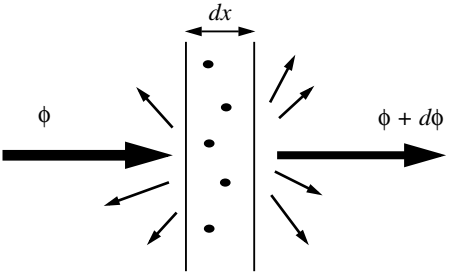


Figure 3.10 Light scattered by an infinitely thin layer containing light-scattering particles at a concentration *c*.

where λ = light wavelength

$\beta(\lambda)$ = scattering coefficient of the medium

By analogy with the absorption phenomenon (see Section 3.4), we introduce the *molar decadic scattering extinction coefficient* (or, in short, *scattering coefficient*) $\sigma(\lambda)$ (unit, $m^2 \cdot mol^{-1}$), which can be interpreted as the scattering cross-section area of a mole of particles of radius *r*.

$$\beta(\lambda) = \sigma(\lambda) c \ln(10) \tag{3.25}$$

$$\sigma(\lambda) = \frac{N_A}{\ln(10)} \pi r^2 \chi_{sc}(\lambda) \tag{3.26}$$

where c = concentration of the particles

$N_A = 6.022 \cdot 10^{23}$ = Avogadro number

$\chi_{sc}(\lambda)$ = scattering efficiency factor of the particle

Note that the total extinction coefficient $\varepsilon_T(\lambda)$ of a particle that has both a scattering and an absorbing behavior corresponds to the sum of the scattering and absorbing cross-section areas (Equations 3.16 and 3.26).

$$\varepsilon_T(\lambda) = \varepsilon(\lambda) + \sigma(\lambda) = [\pi r^2 \chi_{abs}(\lambda) + \pi r^2 \chi_{sc}(\lambda)] \cdot \frac{N_A}{\ln(10)} \tag{3.27}$$

Remark: Sections 3.5.1 and 3.5.2 can be skipped or browsed rapidly for a first reading and revisited later as required.

3.5.1 Rayleigh scattering

The Rayleigh scattering theory applies to independent scattering of particles that are about ten times smaller than the wavelength of the incident light. In 1871, Lord Rayleigh established the following equation, which gives I_θ, the intensity of the light scattered in a direction having an angle θ with the direction of the incident light beam:

$$I_\theta = \left(\frac{\pi^2}{L^2 \varepsilon_0^2 \lambda_0^4}\right) N \alpha^2 \left(\frac{1 + \cos^2\theta}{2}\right) I_0 \tag{3.28}$$

where I_0 = intensity of the incident collimated light beam

λ_0 = its wavelength in empty space

$\varepsilon_0 = 8.842 \cdot 10^{-12}$ = permittivity of empty space (unit, $\mathrm{F \cdot m^{-1}}$)

L = distance at which the intensity I_θ is measured

α = polarizability of the medium

$N = c \cdot N_A$ = number of particles per unit volume

The scattering is rotationally symmetrical about the incident light beam. The detailed calculation can be found in the literature.[14]

The polarizability α of a medium of permittivity $\varepsilon_m = n_m^2 \varepsilon_0$ containing particles of refractive index n and of volume v is given by[15]

$$\alpha = 3\varepsilon_m \cdot \frac{n^2 - n_m^2}{n^2 + 2n_m^2} v \tag{3.29}$$

The scattering coefficient $\beta(\lambda)$ is calculated by integrating Equation 3.28 over a sphere of radius $L = 1$.

$$\beta(\lambda) = \frac{1}{I_0}\oint I_\theta d\omega = \left(\frac{8\pi^3}{3}\frac{1}{\varepsilon_0^2\lambda_0^4}\right)N\alpha^2 = \frac{24\pi^3}{\lambda_m^4}\cdot\left(\frac{n^2-n_m^2}{n^2+2n_m^2}\right)^2 Nv^2 \tag{3.30}$$

Using Equation 3.26, we can also deduce the scattering efficiency factor $\chi_{sc}(\lambda)$.

$$\chi_{sc}(\lambda) = \frac{8}{3}\cdot\left(\frac{n^2-n_m^2}{n^2+2n_m^2}\right)^2\cdot\left(\frac{2\pi r}{\lambda_m}\right)^4 \tag{3.31}$$

where λ_m = wavelength in the medium of refractive index n_m

$\chi_{sc}(\lambda)$ = scattering efficiency factor of the particle

Note that $\beta(\lambda)$ is proportional to λ_m^{-4}, which means that blue light is more strongly scattered than red light. Sunlight scattered in the atmosphere is mostly blue, which explains why the sky is blue. At sunrise and at sunset, the light from the sun has to traverse a thicker atmospheric layer than at noon, so most of the blue light is scattered, and the remaining unscattered light is mostly red.

3.5.2 Mie scattering

The Mie scattering theory[16] is a generalization of the Rayleigh theory, which predicts the scattering behavior of a medium of refractive index n_m containing particles of radius r and of refractive index $\hat{n} = n(1 + i\kappa)$. This theory assumes the absence of multiple scattering. In practice, this means that the distance between two particles is greater than $3r$. The Mie scattering is also rotationally symmetrical about the incident light beam. The intensity I_θ of the light scattered in the direction making an angle θ with the direction of the incident light beam, as derived from Maxwell's equations, is given by[17]

$$I_\theta = \frac{N}{L^2}\left(\frac{\lambda_m}{2\pi}\right)^2\left(\frac{|S_1|^2+|S_2|^2}{2}\right)I_0 \tag{3.32}$$

where λ_m = wavelength of the incident light in the medium of refractive index n_m

N = number of particles per unit of volume

L = distance at which the intensity I_θ is measured

The coefficients S_1 and S_2 are defined by the following series:

$$\begin{cases} S_1 = \sum_{l=1}^{\infty} \frac{2l+1}{l(l+1)} [a_l \cdot \pi_l(\cos\theta) + b_l \cdot \tau_l(\cos\theta)] \\ S_2 = \sum_{l=1}^{\infty} \frac{2l+1}{l(l+1)} [a_l \cdot \tau_l(\cos\theta) + b_l \cdot \pi_l(\cos\theta)] \end{cases} \tag{3.33}$$

The angular functions π_l and τ_l are defined by recurrence as follows:

$$\begin{cases} \pi_l(\cos\theta) = \frac{2l-1}{l-1} \cos\theta \cdot \pi_{l-1}(\cos\theta) - \frac{l}{l-1} \pi_{l-2}(\cos\theta) \\ \tau_l(\cos\theta) = l\cos\theta \cdot \pi_l(\cos\theta) - (l+1)\pi_{l-1}(\cos\theta) \end{cases} \tag{3.34}$$

where the first functions are

$$\begin{aligned} \pi_0(\cos\theta) &= 0 & \pi_1(\cos\theta) &= 1 & \pi_2(\cos\theta) &= 3\cos\theta \\ & & \tau_1(\cos\theta) &= \cos\theta & \tau_2(\cos\theta) &= 3\cos 2\theta \end{aligned} \tag{3.35}$$

The coefficients a_l and b_l are combinations of Bessel and Hankel functions. To simplify their mathematical expressions, let us introduce the following variables:

$$m = \frac{\hat{n}}{n_m}, \quad \gamma = \frac{2\pi r}{\lambda_m}, \quad \text{and} \quad \delta = m\gamma \tag{3.36}$$

Furthermore, let us define the functions Ψ_l and ξ_l.

$$\begin{cases} \Psi_l(x) = \sqrt{\frac{\pi x}{2}} \cdot J_{l+\frac{1}{2}}(x) \\ \xi_l(x) = \sqrt{\frac{\pi x}{2}} \cdot H^{(1)}_{l+\frac{1}{2}}(x) = \sqrt{\frac{\pi x}{2}} \cdot \left[J_{l+\frac{1}{2}}(x) + iY_{l+\frac{1}{2}}(x) \right] \end{cases} \tag{3.37}$$

where $J_{l+\frac{1}{2}}$ = Bessel function of the first kind

$Y_{l+\frac{1}{2}}$ = Bessel function of the second kind

$H^{(1)}_{l+\frac{1}{2}}$ = a Hankel function

The properties of the Legendre polynomial, of the Bessel functions, and of the Hankel function can be found in most handbooks of mathematics.[18] Using Equations 3.36 and 3.37, the coefficients a_l and b_l can be written as follows:

$$\begin{cases} a_l = \dfrac{\Psi_l(\gamma)\Psi_l'(\delta) - m\Psi_l(\delta)\Psi_l'(\gamma)}{\xi_l(\gamma)\Psi_l'(\delta) - m\Psi_l(\delta)\xi_l'(\gamma)} \\ b_l = \dfrac{m\Psi_l(\gamma)\Psi_l'(\delta) - \Psi_l(\delta)\Psi_l'(\gamma)}{m\xi_l(\gamma)\Psi_l'(\delta) - \Psi_l(\delta)\xi_l'(\gamma)} \end{cases} \tag{3.38}$$

Note that S_1 and S_2 are complex if the particles attenuate electromagnetic waves ($\kappa \neq 0$).

For small ($\gamma < 0.8$) and nonattenuating particles ($\kappa = 0$), the series S_1 and S_2 can be limited to the terms a_1, a_2, and b_1.

$$a_1 = \frac{2}{3}\left(\frac{m^2-1}{m^2+2}\right)\gamma^3, \quad a_2 = \frac{-1}{15}\left(\frac{m^2-1}{2m^2+3}\right)\gamma^5, \quad b_1 = -\frac{1}{45}(m^2-1)\gamma^5 \tag{3.39}$$

Note that a_1 corresponds to the contribution of the Rayleigh scattering. If we neglect all terms beyond a_1 in Equation 3.33, we obtain the Rayleigh scattering Equation 3.28.

The scattering coefficient $\sigma(\lambda)$ and the absorption coefficient $\varepsilon(\lambda)$ are obtained by integrating Equation 3.32 over a sphere of radius $L = 1$. This calculation is tedious, as the coefficients a_l and b_l are very complex. Therefore, the scattering efficiency factor $\chi_{sc}(\lambda)$ and the absorption efficiency factor $\chi_{abs}(\lambda)$ for many different values of m and γ have been calculated in the past, and they can be found in tables listed in the literature.[19] Using the approximation for small and nonabsorbing particles (taking into account the terms a_1, a_2, and b_1), the scattering efficiency factor $\chi_{sc}(\lambda)$ of the particle is

$$\chi_{sc}(\lambda) = \frac{8}{3}\gamma^4\left(\frac{m^2-1}{m^2+2}\right)^2\left[1 + \frac{6}{5}\left(\frac{m^2-1}{m^2+2}\right)\gamma^2 + \ldots\right] \tag{3.40}$$

Note that the first term in this series corresponds to the efficiency factor of the Rayleigh scattering that we have already seen in Equation 3.31. For larger particles, further terms in the series S_1 and S_2 must be taken into account. Nowadays, computers allow us to calculate all coefficients numerically.

It is found that scattering is proportional to λ_m^{-4} in the Rayleigh region ($\gamma \ll 1$) and that, with further increase of γ, it tends to become proportional to λ_m^0, i.e., wavelength independent. Therefore, the light scattered by large particles (e.g., smoke particles) is white. Furthermore, the forward scattering becomes greater than the backward scattering with increasing γ. Figure 3.11 shows the scattering diagram of spherical gold particles for different radii. The Mie theory successfully predicts the spectra of colloidal suspensions, metallic suspensions, and atmospheric dust.

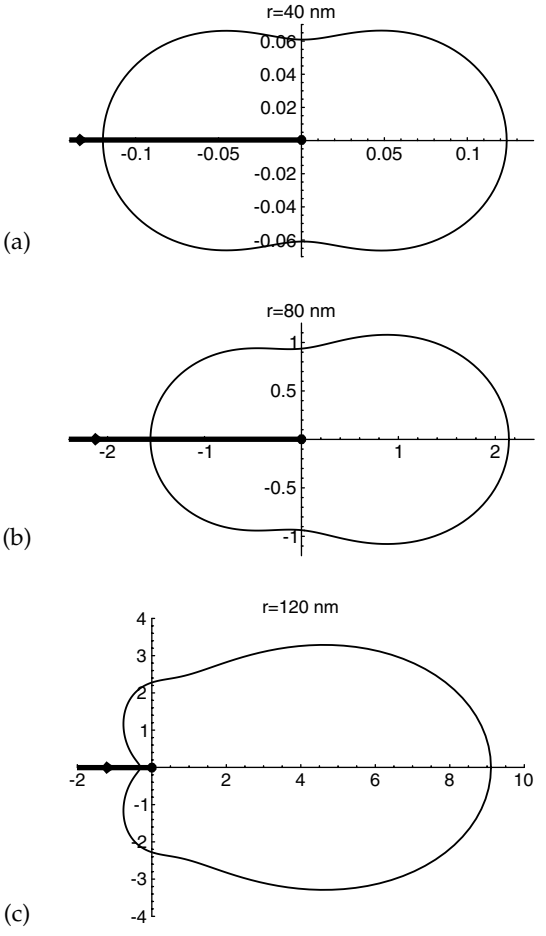


Figure 3.11 Scattering diagram according to Mie for spherical gold particles (λ_m = 550 nm, n_m = 1.33, $\hat{n}$ = 0.57 + i2.45) with radii (a) r = 40 nm ($\gamma \ll 1$ Rayleigh region), (b) r = 80 nm ($\gamma \approx 1$), and (c) r = 120 nm ($\gamma > 1$). The unit of the axes is $(\lambda_m/(2\pi))^2$.

3.5.3 Multiple scattering

In the framework of the Rayleigh and the Mie theory, we assumed that the scattering particles are independent; i.e., the light scattered by one particle does not interact with other particles. With diminishing distance between the particles or increasing thickness of the medium, this assumption no longer holds, and single scattering gives way to multiple scattering. It can be shown that, for a sufficient number of particles, regardless of the scattering law used, an isotropic distribution ultimately arises.[20] With multiple scattering, the characteristic properties of single scattering disappear more or less rapidly according to the given conditions.

3.6 Phenomenological models

There is no general quantitative solution to the problem of multiple scattering for large particles ($2\pi r/\lambda_m \geq 1$) that are tightly packed. In such cases, phase relations and interferences arise among the scattered beams. Therefore, new approaches based on phenomenological theories had to be developed. Throughout Section 3.6, the wavelength designation (λ) is dropped to simplify the notation, but I, ρ, ε_T, i, j, K, S, ρ_g, R_g, and R are functions of wavelength.

3.6.1 Radiative transfer

The astronomer S. Chandrasekhar established in 1947 the radiative transfer equation,[21] which corresponds to an energy balance. This equation describes the intensity change dI of a light beam of given wavelength along a path of length ds within a medium of density ρ and total extinction coefficient ε_T.

$$\frac{dI}{ds} = -\ln(10)\varepsilon_T \rho I + \rho j \tag{3.41}$$

where $j/(\ln(10)\varepsilon_T)$ corresponds to the *source function* characterizing a light source. In the particular case of a nonluminescent medium, j is a *scattering function* defined as

$$j(\theta, \varphi) = \frac{\ln(10)\varepsilon_T}{4\pi}\int_0^{\pi}\int_0^{2\pi} p(\theta, \varphi;\theta', \varphi')I(\theta', \varphi')\sin\theta' d\theta' d\varphi' \tag{3.42}$$

The function $p(\theta, \varphi;\theta', \varphi')$ is called the *phase function*. It gives the amount of intensity that is scattered into a solid angle $d\omega$ of direction (θ, φ) if a beam of radiation in the solid angle $d\omega' = \sin\theta' d\theta' d\varphi'$ strikes a mass element of the medium (see Figure 3.12). Note that the phase function is normalized as follows:

$$\frac{1}{4\pi}\int\int p(\theta, \varphi;\theta', \varphi')\sin\theta d\theta d\varphi = \varpi_0 \leq 1 \tag{3.43}$$

where ϖ_0 = fraction of light lost from an incident beam due to scattering

Term ϖ_0 is the albedo of the medium. The simplest example of phase function is $p(\theta, \varphi;\theta', \varphi') = 1$ in the case of isotropic scattering. Another case of interest is Rayleigh's phase function, which corresponds to the angular term of Equation 3.28.

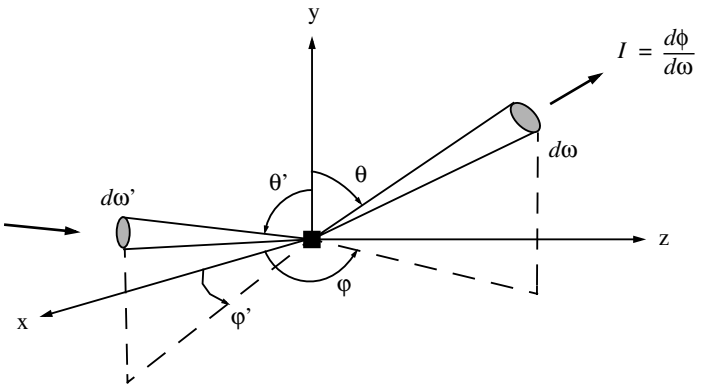


Figure 3.12 The phase function gives the amount of intensity that is scattered into a solid angle $d\omega$ of direction (θ, φ) if a beam of radiation in the solid angle $d\omega' = \sin\theta' d\theta' d\varphi'$ strikes a mass element of the medium.

$$p(\cos\Theta) = \frac{3}{2}\left(\frac{1 + \cos^2\Theta}{2}\right) \tag{3.44}$$

where Θ is the angle between the direction of the incident radiation in $d\omega'$ and the direction of the solid angle $d\omega$. In general, the phase function can be expanded as a series in Legendre polynomials of the form

$$p(\cos\Theta) = \sum_{l=0}^{\infty} \varpi_l P_l(\cos\Theta) \tag{3.45}$$

where P_l = the Legendre polynomial of degree l

ϖ_l = a constant

Combining Equations 3.41 and 3.42 leads to an integro-differential equation that is difficult to solve. Nevertheless, solutions could be calculated in a few particular cases, such as for isotropic scattering in a medium made of parallel planes.[21]

The radiative transfer equation is very powerful, but it requires a tedious mathematical treatment. Therefore, simplified versions of this theory are used in practice, e.g., the Kubelka–Munk model and the multichannel model, which are presented in the following sections.

3.6.2 Kubelka–Munk model (two-flux model)

Let us consider a reflector made of a reflecting substrate of reflectance ρ_g in optical contact with a light-absorbing and light-scattering medium of thick-

ness X (see Figure 3.13). The scattering is assumed to have an isotropic distribution, as it results from multiple scattering (see Section 3.5.3). In 1931, Kubelka and Munk[22] proposed a reflection model based on two diffuse light fluxes: $i(x)$ oriented downward and $j(x)$ oriented upward.

Let us analyze the variation of these fluxes when they cross a layer of infinitesimal thickness dx. The x axis is oriented upward, and the origin is set at the top of the substrate. Let K be the phenomenological absorption coefficient corresponding to the fraction of the light flux absorbed by the infinitesimal layer. Let S be the phenomenological scattering coefficient corresponding to the fraction of the light flux that is scattered backward by the infinitesimal layer.

We first analyze the variation of $j(x)$ when it crosses the layer. The flux $j(x)$ is reduced due to absorption within the infinitesimal layer by an amount $Kj(x)dx$, and the backscattering further reduces the flux by an amount $Sj(x)dx$. However, the flux $j(x)$ is increased by the light that is backscattered when the flux $i(x)$ crosses the same layer: $Si(x)\,dx$. Putting these elements together leads to the following equation:

$$\frac{dj(x)}{dx} = -(K+S)j(x) + Si(x) \tag{3.46}$$

The same analysis performed for the flux $i(x)$ leads to a similar relation (notice the orientations along the vertical x-axis).

$$\frac{di(x)}{dx} = (K+S)i(x) - Sj(x) \tag{3.47}$$

Note that, in a transparent medium, S equals 0, and differential Equations 3.46 and 3.47 lead to Beer's law for diffuse light (Equation 3.23).

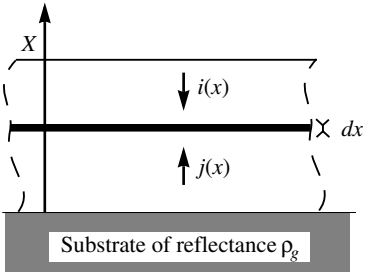


Figure 3.13 Light-absorbing and light-scattering medium of thickness X that is in optical contact with a substrate of reflectance ρ_g. The medium is divided into parallel layers of infinitesimal thickness *dx*. Note that two fluxes are considered: *i(x),* which is oriented downward, and *j(x)*, which is oriented upward.

Equations 3.46 and 3.47 together form a system of linear differential equations that describes the variation of $i(x)$ and $j(x)$ when they cross an infinitesimal layer of thickness dx.

$$\begin{cases} \dfrac{di(x)}{dx} = (K+S)i(x) - Sj(x) \\ \dfrac{dj(x)}{dx} = -(K+S)j(x) + Si(x) \end{cases} \tag{3.48}$$

Kubelka and Munk solved Equation 3.48 using a traditional calculation method.[22] Here, we propose a more modern approach based on matrix algebra. The system in Equation 3.48 can be written in matrix form as follows:

$$\begin{bmatrix} \dfrac{di(x)}{dx} \\ \dfrac{dj(x)}{dx} \end{bmatrix} = \begin{bmatrix} K+S & -S \\ S & -(K+S) \end{bmatrix} \cdot \begin{bmatrix} i(x) \\ j(x) \end{bmatrix} \tag{3.49}$$

This kind of matrix differential equation has a well-known solution, which is given by the exponential of the matrix.[23] By integrating the equation between $x = 0$ and $x = X$, we obtain

$$\begin{aligned} \begin{bmatrix} i(X) \\ j(X) \end{bmatrix} &= \exp\left(\begin{bmatrix} K+S & -S \\ S & -(K+S) \end{bmatrix} (X-0) \right) \cdot \begin{bmatrix} i(0) \\ j(0) \end{bmatrix} \\ &= \begin{bmatrix} t & u \\ v & w \end{bmatrix} \cdot \begin{bmatrix} i(0) \\ j(0) \end{bmatrix} \end{aligned} \tag{3.50}$$

where t, u, v, w = elements of the matrix exponential

$i(0)$, $j(0)$ = intensities of the fluxes i and j at $x = 0$

Note that the exponential of a matrix M is defined by the following power series:

$$\exp(\mathrm{M}) = \sum_{l=0}^{\infty} \frac{(\mathrm{M})^l}{l!} \tag{3.51}$$

From Equation 3.50 and the boundary condition $j(0) = \rho_g \cdot i(0)$, we can derive by algebraic manipulations[24,25] all the well-known results of the Kubelka–Munk theory that are listed in the literature.[26] The most important result is the hyperbolic solution of the Kubelka–Munk model,

$$\rho = \frac{j(X)}{i(X)} = \frac{v + \rho_g \cdot w}{t + \rho_g \cdot u} = \frac{1 - \rho_g \cdot (a - b \cdot \coth(bSX))}{a - \rho_g + b \cdot \coth(bSX)} \tag{3.52}$$

where $a = (S + K)/S$ and $b = \sqrt{a^2 - 1}$. The ratio $\rho = j(X)/i(X)$ is called the *body (or true) reflectance*[27] of the analyzed sample. To characterize the medium alone, practitioners use the reflectance $\rho_\infty = j(X_\infty)/i(X_\infty)$ of a medium that is so thick that further increase in thickness fails to change its reflectance. In other words, if an additional layer of thickness X is put on top of such a medium, we have $\rho_\infty = j(X + X_\infty)/i(X + X_\infty)$. According to Equation 3.50, we have

$$\begin{bmatrix} i(X + X_\infty) \\ j(X + X_\infty) \end{bmatrix} = \begin{bmatrix} t & u \\ v & w \end{bmatrix} \cdot \begin{bmatrix} i(X_\infty) \\ j(X_\infty) \end{bmatrix}$$
$$i(X + X_\infty) \begin{bmatrix} 1 \\ \rho_\infty \end{bmatrix} = \begin{bmatrix} t & u \\ v & w \end{bmatrix} \cdot i(X_\infty) \begin{bmatrix} 1 \\ \rho_\infty \end{bmatrix} \tag{3.53}$$

From a mathematical point of view, this means that the vector $[1, \rho_\infty]$ is an eigenvector[28] of the matrix given in Equation 3.50, and that $\alpha = i(X + X_\infty)/i(X_\infty)$ is the corresponding eigenvalue. This observation permits us to obtain ρ_∞ by calculating the eigenvectors of this matrix. By solving the characteristic polynomial of the matrix,[29] it can be shown that this matrix has two eigenvalues,

$$\alpha_1 = \frac{(w + t) - \sqrt{(w - t)^2 + 4uv}}{2} = -\sqrt{K^2 + 2KS}$$
$$\alpha_2 = \frac{(w + t) + \sqrt{(w - t)^2 + 4uv}}{2} = \sqrt{K^2 + 2KS} \tag{3.54}$$

that are associated with the following eigenvectors, respectively:

$$\mathbf{V}_1 = \begin{bmatrix} 1 \\ \dfrac{(w - t) - \sqrt{(w - t)^2 + 4uv}}{2u} \end{bmatrix} \qquad \mathbf{V}_2 = \begin{bmatrix} 1 \\ \dfrac{(w - t) + \sqrt{(w - t)^2 + 4uv}}{2u} \end{bmatrix} \tag{3.55}$$

Being a reflectance value, ρ_∞ must be in the range between 0 and 1. But because the second component of $\mathbf{V}_1$ is outside the range $[0, 1]$, the solution given by $\mathbf{V}_1$ must be discarded, and ρ_∞ is simply the second component of the eigenvector $\mathbf{V}_2$.

$$\rho_\infty = \frac{(w-t)+\sqrt{(w-t)^2+4uv}}{2u} = 1+\frac{K}{S}-\sqrt{\frac{K^2}{S^2}+2\frac{K}{S}} \tag{3.56}$$

This result is often presented in a more compact form known as the *Kubelka–Munk function*.

$$\frac{K}{S} = \frac{(1-\rho_\infty)^2}{2\rho_\infty} \tag{3.57}$$

Other important results are the reflectance ρ_0 of a layer with ideal black background ($\rho_g = 0$), and the reflectance ρ_1 of a layer with ideal white background ($\rho_g = 1$).

$$\rho_0 = \frac{v}{t} = \frac{1}{a+b\coth(bSX)} \tag{3.58}$$

$$\rho_1 = \frac{v+w}{t+u} = \frac{1-(a-b\coth(bSX))}{a+b\coth(bSX)} \tag{3.59}$$

Traditionally, ρ_0, ρ_1, and the thickness X of the medium are used to determine the coefficients K and S. In a first step, we extract a from Equations 3.58 and 3.59.

$$a = \frac{1}{2}\left(1-\frac{\rho_1-1}{\rho_0}\right) \tag{3.60}$$

Once a is known, $b = \sqrt{a^2-1}$ is obtained immediately. In a second step, we extract S from Equation 3.58.

$$S = \frac{1}{bX}\operatorname{acoth}\frac{1-a\rho_0}{b\rho_0} \tag{3.61}$$

Finally, from the definition of a,

$$K = S(a-1) \tag{3.62}$$

Note that the phenomenological coefficients K and S can be related to the fundamental optical properties introduced previously. In Section 3.4, we generalized Beer's law for diffuse light (see Equation 3.23), which is equivalent to $d\phi = -K\phi dx$. Therefore, we have (see Equation 3.16)

$$K = 2\ln 10 \cdot \varepsilon(\lambda) \cdot c = 2N_A \cdot c \cdot \pi r^2 \cdot \chi_{abs}(\lambda) \tag{3.63}$$

A similar calculation allows us to relate S to the scattering coefficient $\sigma(\lambda)$. Because the scattering in the medium is assumed to have an isotropic distribution, the scattering coefficient must be divided by two, because S accounts only for backward-scattered light. Hence, we obtain from Equations 3.25 and 3.26,

$$S = 2\ln 10 \cdot \frac{\sigma(\lambda)}{2} \cdot c = N_A \cdot c \cdot \pi r^2 \cdot \chi_{sc}(\lambda) \tag{3.64}$$

3.6.3 *Surface phenomena and Saunderson correction*

In the Kubelka–Munk theory, the diffuse reflector is modeled by a light-absorbing and light-scattering medium in optical contact with a substrate that is supposed to be a Lambertian[30] reflector of reflectance ρ_g. In a medium having a refractive index n different from that of air, surface reflection and multiple internal reflections occur[31] as shown in Figure 3.14. As a consequence, the reflectances prevailing in a medium of refractive index n can differ greatly from the reflectances measured at its surface. Traditionally, this is taken into account by applying the Saunderson correction[32] to the computed spectrum. In this section, we write the Saunderson correction in matrix form, to be applied to Equation 3.50.

Let us denote by i the incident flux on the external surface of the paper and by j the flux emerging from the paper. Let r_s be the fraction of diffuse light reflected by the air-medium interface (external surface of the reflector), and let r_i be the fraction of diffuse light reflected internally by the air–medium interface (internal surface of the medium); see Figure 3.15. According to Equations 3.13 and 3.14, the values of r_s and r_i depend only on the refractive index n of the medium. Judd[11] has computed their numerical values for a large number of refractive indices (see Table 3.1).

The balance of the fluxes at the air–medium interface, as shown in Figure 3.15, leads to the following system of equations for *i(X)*, the incident flux below the air–medium interface, and for *j*, the emerging flux above the air–medium interface:

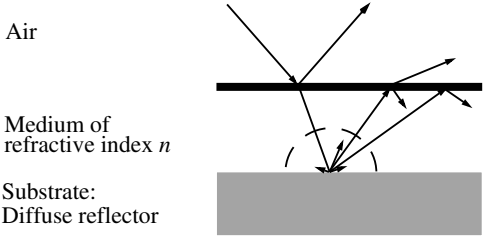


Figure 3.14 Surface reflection and multiple internal reflections caused by the interface between the air and the medium.

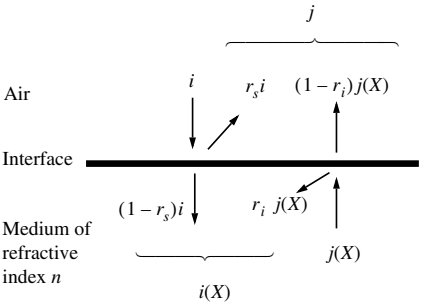


Figure 3.15 External and internal reflections of the upward and downward fluxes on the air–medium interface.

$$\begin{cases} i(X) = (1 - r_s) i + r_i j(X) \\ j = r_s i + (1 - r_i) j(X) \end{cases} \tag{3.65}$$

Assuming that the refractive index of the medium is constant over the whole visible range of wavelengths, r_s and r_i are also constant. Hence, Equation 3.65 can be written in the following matrix form:

$$\begin{bmatrix} i \\ j \end{bmatrix} = \begin{bmatrix} \dfrac{1}{1 - r_s} & \dfrac{-r_i}{1 - r_s} \\ \dfrac{r_s}{1 - r_s} & \left(1 - r_i - \dfrac{r_s r_i}{1 - r_s}\right) \end{bmatrix} \begin{bmatrix} i(X) \\ j(X) \end{bmatrix} \tag{3.66}$$

We call the matrix in Equation 3.66 the Saunderson correction matrix. Note that this correction matrix can be generalized for any interface between a medium of refractive index n_1 and a medium of refractive index n_2. According to Section 3.3.2, the values of r_s and r_i are then given by

$$r_s = \int_0^{\frac{\pi}{2}} (r_{n_1, n_2}(\theta) \cdot \sin 2\theta) d\theta \text{ and } r_i = \int_0^{\frac{\pi}{2}} (r_{n_2, n_1}(\theta) \cdot \sin 2\theta) d\theta \tag{3.67}$$

The Saunderson correction is obtained by combining Equations 3.66 and 3.50.

$$\begin{bmatrix} i \\ j \end{bmatrix} = \begin{bmatrix} \dfrac{1}{1 - r_s} & \dfrac{-r_i}{1 - r_s} \\ \dfrac{r_s}{1 - r_s} & \left(1 - r_i - \dfrac{r_s r_i}{1 - r_s}\right) \end{bmatrix} \cdot \exp\left(\begin{bmatrix} K + S & -S \\ S & -(K + S) \end{bmatrix} X \right) \cdot \begin{bmatrix} i(0) \\ j(0) \end{bmatrix} = \begin{bmatrix} t' & u' \\ v' & w' \end{bmatrix} \cdot \begin{bmatrix} i(0) \\ j(0) \end{bmatrix} \tag{3.68}$$

We denote the elements of the product matrix by t', u', v', and w'. These coefficients and the boundary condition $j(0) = \rho_g \cdot i(0)$ allow the calculation of the reflectance R.

$$R = \frac{j}{i} = \frac{v' + \rho_g \cdot w'}{t' + \rho_g \cdot u'} \tag{3.69}$$

This equation allows us to compute the reflectance under diffuse light illumination of a light-absorbing and light-scattering medium in optical contact with a substrate of known reflectance ρ_g. If we develop the product in Equation 3.68 algebraically, we obtain the famous Saunderson corrected reflection formula.[33]

$$R = r_s + \frac{(1 - r_s)(1 - r_i)\rho}{1 - r_i\rho} \tag{3.70}$$

where $\rho = j(X)/i(X)$ = the body reflectance given by the Kubelka–Munk model

Figure 3.16 shows the reflectance and the body reflectance of a cyan sample.

In the graphic arts, most measuring instruments use a 45°/0° measuring geometry wherein the incident light beam is collimated with an incidence of 45°, and the detector is placed at an angle of 0° (see Figure 3.17). This set-up prevents the light reflected specularly from entering the detector, hence $r_s = 0$. Furthermore, in the particular case of a nonscattering medium ($S = 0$), the refracted light beam, with the normal to the surface, forms an angle of $\alpha = \text{asin}[1/(n\sqrt{2})]$, where n is the refractive index of the medium. The entering collimated light beam follows within the medium a path of length $X/(\cos\alpha)$, which is shorter than the average path of length $2X$ followed by diffuse light (see Section 3.4). Because the detector is at an angle of 0°, only the light emerging with an angle of 0° is detected. This emerging light beam follows in the medium a path of length X, which is also shorter

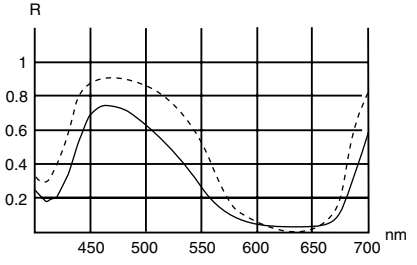


Figure 3.16 Body reflection spectrum (dashed line) and reflection spectrum (continuous line) of a cyan sample.

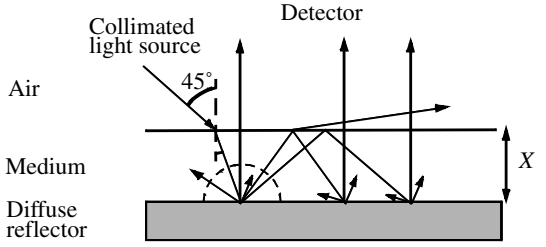


Figure 3.17 Path followed in the medium by the collimated light beam produced by a measuring instrument having a 45°/0° measuring geometry.

than the average path length of $2X$ followed by diffuse light. Because the total path length of the light beam within the medium is shorter, the absorption of the light beam within the medium is not the same as for diffuse light. Therefore, the Saunderson correction matrix in Equation 3.66 must be modified as follows so as to take the 45°/0° geometry into account:[34]

$$\begin{bmatrix} i \\ j \end{bmatrix} = \begin{bmatrix} \exp\left[\left(\frac{1}{2\cos\alpha} - 1\right) \cdot KX\right] & -r_i \exp\left[\left(\frac{1}{2\cos\alpha} - 1\right) \cdot KX\right] \\ 0 & (1 - r_i) \exp\left[\frac{KX}{2}\right] \end{bmatrix} \begin{bmatrix} i(X) \\ j(X) \end{bmatrix} \tag{3.71}$$

Considering the particular case of a medium with refractive index $n = 1.5$, we have $(1/(2\cos\alpha) - 1) = -0.44$. The modified Saunderson correction matrix leads, after developing Equation 3.69, to the Williams–Clapper equation,[35]

$$R = \frac{(1 - r_i)\rho_g \exp[-1.06KX]}{1 - r_i \rho_g \exp[-2KX]} \tag{3.72}$$

Note that ρ_g is the reflectance of the substrate within the medium of refractive index n. The surface phenomena do not allow us to measure ρ_g directly. Let R_g be the reflectance of the substrate measured in air without the medium on top of it. If the substrate has the same refractive index n as the medium, ρ_g is deduced from R_g by deriving the following formula from Equation 3.70:

$$\rho_g = \frac{1}{r_i + \dfrac{(1 - r_s)(1 - r_i)}{R_g - r_s}} \tag{3.73}$$

In practice, the substrate is not always available without the coating medium on top of it. This happens, for example, in the case of the high-

quality paper used in graphic arts, where the fiber substrate is coated with an ink-absorbing layer of refractive index n. Assuming that the coating is transparent, we can deduce ρ_g from Equation 3.73 by replacing R_g with the measured reflectance of the paper.

Note that the matrix formulation of Equation 3.68 gives a better overview of the modeled system. Instead of using several functions nested within each other, the analyzed sample is simply modeled by the product of two matrices.

3.6.4 *Multichannel model*

The two-flux model proposed by Kubelka and Munk corresponds to a significant simplification of the radiative transfer equation (see Section 3.6.1). To improve the quality of the prediction, Mudget and Richards in 1971 proposed an intermediate model by considering a larger number of light fluxes.[36] Each flux propagates in a different fraction of space called a *channel* (see Figure 3.18). Therefore, this theory is called the *multichannel model* or the *multiple flux theory*. In this context, the radiative transfer equation corresponds to a model that considers an infinite number of fluxes.

The multichannel model considers $2m$ fluxes; m fluxes denoted j_l are oriented upward, and m fluxes denoted i_l are oriented downward. Let us denote K_p as the absorption coefficient in the p th channel, and $S_{p,l}$ as the scattering coefficient from the pth channel into the lth channel. The scattering coefficients $S_{p,l}$ are computed by using a scattering model, as for instance the Mie model presented in Section 3.5.2.

As in the Kubelka–Munk model, we analyze the variation of each flux when it is traversing an infinitesimal layer of thickness dx. This gives us $2m$ linear differential equations of the first order with $2m$ variables. These equations can be written in matrix form as follows:

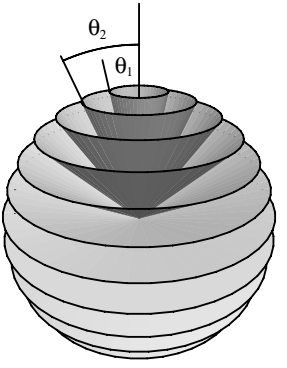


Figure 3.18 In the multichannel model the whole space is subdivided into $2m$ channels where the lth channel corresponds to the space between the cone of angle θ_{l-1} and the cone of angle θ_l.

$$\frac{d}{dx}\begin{bmatrix} i_1 \\ \vdots \\ i_m \\ j_{m+1} \\ \vdots \\ j_{2m} \end{bmatrix} \tag{3.74}$$

$$= \begin{bmatrix} \left(K_1 + \sum_{l \neq 1} S_{1,l}\right) \cdots & -S_{m,1} & -S_{m+1,1} & \cdots & -S_{2m,1} \\ \vdots & \vdots & \vdots & & \vdots \\ -S_{1,m} & \cdots \left(K_m + \sum_{l \neq m} S_{m,l}\right) & -S_{m+1,m} & \cdots & -S_{2m,m} \\ S_{1,m+1} & \cdots S_{m,m+1} & -\left(K_{m+1} + \sum_{l \neq m+1} S_{m+1,l}\right) & \cdots & S_{2m,m+1} \\ \vdots & \vdots & \vdots & & \vdots \\ S_{1,2m} & \cdots S_{m,2m} & S_{m+1,2m} & \cdots & -\left(K_{2m} + \sum_{l \neq 2m} S_{2m,l}\right) \end{bmatrix}$$

$$\cdot \begin{bmatrix} i_1 \\ \vdots \\ i_m \\ j_{m+1} \\ \vdots \\ j_{2m} \end{bmatrix}$$

The sign inversion in the first m rows occurs because the x axis is oriented upward. The pth element on the diagonal of the matrix corresponds to the attenuation of the pth light flux. This attenuation is caused by the light absorption K_p in the pth channel and by the scattering of light from the pth channel into all other channels,

$$\sum_{l \neq p} S_{p,l}$$

The off-diagonal element $S_{p,l}$ corresponds to the light received by the flux l from the channel p.

The solution of Equation 3.74 is also given by the exponential of the matrix. As in the Kubelka–Munk model, the boundary conditions at the surface of the substrate define the relations between the fluxes $j_l(0)$ and the fluxes $i_p(0)$. An extended Saunderson correction matrix that allows us to

predict the reflectance with a higher accuracy can be also defined. The complete treatment is beyond the scope of this chapter, but the mathematical procedure is the same as presented in the Sections 3.6.2 and 3.6.3. Note that practitioners in the paint industry normally use the four-flux theory.[37]

3.7 *The fluorescence phenomenon*

Let us first recall the basic principles of molecular fluorescence.[38] We consider a theoretical molecule having two electronic energy states, E_0 (ground state) and E_1 (excited state). Each electronic state has several vibrational states (see Figure 3.19). Incident polychromatic light (photons) excites the molecules that are in state E_0 and makes them temporarily populate the excited vibrational states of E_1 (Figure 3.19a).

A vibrational excited state has an average lifetime of only 10^{-15} s. The molecule rapidly loses its vibrational energy and goes down to the electronic energy state E_1. This relaxation process is nonradiative, and it is caused by the collisions with other molecules to which the vibrational energy is transferred. This induces a slight increase of the temperature of the medium. The excited state E_1 has a lifetime varying between 10^{-6} and 10^{-9} s. Now, there are two ways for the molecule to give up its excess energy. One of them is called *internal conversion*, a nonradiative relaxation for which the mechanism is not fully understood. The transition occurs between E_1 and the upper vibrational state of E_0 (Figure 3.19b), and the lost energy raises the temperature of the medium. The other possible relaxation process is fluorescence. It takes place by emitting a photon of energy corresponding to the transition between E_1 and a vibrational state of E_0 (Figure 3.19c). The remaining excess energy with respect to E_0 is lost by vibrational relaxation. To quantify the number of photons emitted by fluorescence, the *quantum yield* is introduced as the rate of absorbed photons that are released by radiative relaxation.

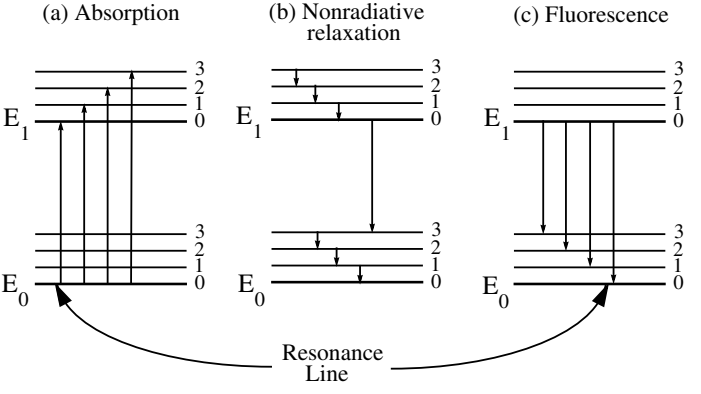


Figure 3.19 The energy level diagram of (a) absorption, (b) nonradiative relaxation, and (c) fluorescent emission.

The wavelength band of absorbed radiation that is responsible for the excitation of the molecules is called the *excitation spectrum*. This spectrum consists of lines whose wavelengths correspond to the energy differences between excited vibrational states of E_1 and the ground electronic state E_0 (according to the energy difference ΔE produced by the absorption of a photon of wavelength λ: $\Delta E = (hc)/\lambda$, where h is Planck's constant and c is the speed of light). The *fluorescence emission spectrum* (or *fluorescence spectrum*), on its part, consists of lines that correspond to the energy differences between the electronic level E_1 and the vibrational states of E_0. The multitude of lines in both spectra is difficult to resolve and makes them look like continuous spectra. Note that the fluorescence spectrum is made up of lines of lower energy than the absorption spectrum. This wavelength shift between the absorption band and the fluorescence band is called the *Stokes shift*. A particular case in which the absorbed photon has the same energy as the one re-emitted by fluorescence is called the *resonance line*.

The shape of the fluorescence emission spectrum does not depend on the spectrum of the absorbed light, but on the probability of the transition between the excited state E_1 and the vibrational states of E_0. Often, the fluorescence spectrum looks like a mirror image of the excitation spectrum (Figure 3.25);[39] this is due to the fact that the differences between vibrational states are about the same in ground and excited states.

Experience shows that fluorescence is favored in rigid molecules that contain aromatic rings.[40] This can easily be understood, as a rigid molecule has a lower possibility of relaxing by a nonradiative process. In fact, the lower the probability of nonradiative relaxation, the higher the quantum yield. Hence, a rise in the medium's viscosity induces a higher fluorescence. In the particular case of inks, the liquid substance fluoresces less than the dried-up printed ink, whose molecules have less degrees of freedom. On the other hand, a rise of the ambient temperature implies a higher probability of nonradiative relaxation due to collisions with other molecules, and a drop in fluorescence is observed.

The fluorescence spectrum is measured with a fluorescence spectrometer.[41] A sample of the unknown fluorescent substance is excited with a monochromatic light beam whose wavelength is within the excitation band of the molecule. The emitted light is analyzed, and the resulting spectrum is the fluorescence spectrum. Its amplitude is maximal when the wavelength of the incident light corresponds to the maximum absorption of the fluorescent molecule. We denote by $f(\lambda)$ the normalized fluorescence spectrum whose integral equals 1. A method for determining the quantum yield is described in Section 3.7.4.

At high concentrations, the behavior of the fluorescent substance is no longer linear. The absorption is too large, and no light can pass through to cause excitation. Temperature, dissolved oxygen, and impurities reduce the quantum yield; therefore, they also reduce the fluorescence. This phenomenon is called *quenching*. In our model, we will suppose that no quenching occurs.

3.7.1 *Fluorescence: transparent layer*

To establish a mathematical formula that predicts the behavior of a transparent medium containing fluorescent molecules, we consider a slice of thickness dx. We denote by $\varepsilon(\lambda)$ the absorption coefficient of the fluorescent molecules, by c their concentration, and by Q their quantum yield in this medium. In the model for transparent media,[42] only the positive direction of propagation is taken into account (see Figure 3.20).

The intensity variation $d\phi$ of the light emerging in the positive direction has two components. The first, $d\phi_1(\lambda)$, is due to the light that has been absorbed. As we have already seen in Equation 3.23, for diffuse light, this absorption is twice[43] the value given by Beer's law.

$$d\phi_1(\lambda, x) = -2\ln 10\, c\varepsilon(\lambda)\phi(\lambda, x)dx \tag{3.75}$$

The second component, $d\phi_2(\lambda, x)$, is the light emitted by fluorescence. The fluorescent molecules emit a fraction Q of the photons absorbed in the excitation band Δ and spread them over the whole emission band defined by the normalized fluorescence spectrum $f(\lambda)$. Due to the fact that fluorescent emission is made in all directions of space, only one-half of the photons go into the positive direction. Hence, the quantum yield must be divided by two. The second component $d\phi_2(\lambda, x)$ is therefore given by

$$d\phi_2(\lambda, x) = 2\ln 10\, c\frac{Q}{2} \cdot f(\lambda)\left[\int_{\Delta}\varepsilon(\mu)\phi(\mu, x)d\mu\right]dx \tag{3.76}$$

The integral between square brackets multiplied by dx equals the amount of absorbed energy. Equation 3.76 leads to the following differential form, which is an extension of Beer's law for diffuse light and fluorescent media:

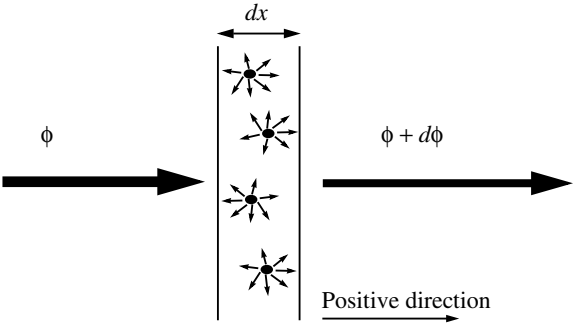


Figure 3.20 Absorption and emission in an infinitely thin fluorescent layer which is irradiated by a diffuse light flux ϕ.

$$d\phi(\lambda, x) = -2\ln 10\, c\varepsilon(\lambda)\phi(\lambda, x)dx + 2\ln 10\, c\frac{Q}{2}\cdot f(\lambda)\left[\int_{\Delta}\varepsilon(\mu)\phi(\mu, x)d\mu\right]dx \tag{3.77}$$

This can be simplified due to the fact that we work with a finite number of wavelength bands whose widths are $\Delta\lambda$, so the integral is replaced by a finite sum. The new relation is given in Equation 3.78, where the index i runs through the wavelength bands.

$$d\phi(\lambda_i, q) = -2\ln 10\, c\,\varepsilon(\lambda)\phi(\lambda_i, x)dx + 2\ln 10\, c\,\frac{Q}{2}\cdot f(\lambda_i)\left[\sum_{j\,\in\,\Delta}\varepsilon(\lambda_j)\phi(\lambda_j, x)\Delta\lambda\right]dx \tag{3.78}$$

Writing Equation 3.78 for each of the bands leads to a system of linear differential equations with constant coefficients that can be put into matrix form. If we denote $F_{i,j} = \varepsilon(\lambda_j)f(\lambda_i)\Delta\lambda Q/2$, we obtain Equation 3.79.

$$\begin{bmatrix} \dfrac{d\phi(\lambda_1, x)}{dx} \\ \vdots \\ \dfrac{d\phi(\lambda_i, x)}{dx} \\ \vdots \\ \dfrac{d\phi(\lambda_n, x)}{dx} \end{bmatrix} = -2c\ln 10 \begin{bmatrix} \varepsilon(\lambda_1) & 0 & \cdots & & & & 0 \\ -F_{2,1} & \ddots & & & & & \vdots \\ \vdots & & \ddots & & & & \vdots \\ -F_{i,1} & \cdots & -F_{i,j} & \cdots & \varepsilon(\lambda_i) & & \vdots \\ \vdots & & \vdots & & & \ddots & \vdots \\ -F_{n,1} & \cdots & -F_{n,j} & \cdots & & -F_{n,n-1} & \varepsilon(\lambda_n) \end{bmatrix} \cdot \begin{bmatrix} \phi(\lambda_1, x) \\ \vdots \\ \phi(\lambda_j, x) \\ \vdots \\ \phi(\lambda_n, x) \end{bmatrix} \tag{3.79}$$

The fact that the emitted photon has less energy than the absorbed one implies that $F_{i,j} = 0$ for $\lambda_j \geq \lambda_i$; hence the matrix is triangular.

The solution of equations such as Equation 3.79 has already been investigated by mathematicians.[23] Systems of differential equations whose general expression is $d\Phi/dx = -c\mathrm{M} \cdot \Phi$ (where M is the constant square matrix of Equation 3.79 and Φ is the column vector containing $\phi(\lambda_1, x), \ldots, \phi(\lambda_n, x)$) admit as a solution, when x is integrated between 0 and X,

$$\Phi(X) = \exp(-\mathrm{M}cX) \cdot \Phi(0) \tag{3.80}$$

The vector $\Phi(0)$ is the spectrum of the incident light (light source), and $\Phi(X)$ is the spectrum of the light emerging from a slice of thickness X of the fluorescent medium. The exponential of the matrix $-\mathrm{M}cX$ is defined as follows:

$$\exp(-\mathrm{M}cX) = \sum_{i=0}^{\infty} \frac{(-\mathrm{M}cX)^i}{i!} \tag{3.81}$$

where

$$\mathrm{M} = 2\ln 10 \begin{bmatrix} \varepsilon(\lambda_1) & 0 & \cdots & 0 \\ -F_{2,1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ -F_{i,1} & \cdots & -F_{i,j} & \varepsilon(\lambda_i) & \cdots & \vdots \\ \vdots & & \vdots & & \ddots & \vdots \\ \vdots & & \vdots & & & 0 \\ -F_{n,1} & \cdots & -F_{n,j} & \cdots & -F_{n,n-1} & \varepsilon(\lambda_n) \end{bmatrix} \tag{3.82}$$

We will call M the *fluorescence density matrix*. The transmission spectrum $T(\lambda)$ resulting from the combined action of fluorescence and absorption can be computed for each wavelength λ using the expression $T(\lambda) = \phi(\lambda, X)/\phi(\lambda, 0)$, where $\phi(\lambda, X)$ and $\phi(\lambda, 0)$ are, respectively, components of $\Phi(X)$ and $\Phi(0)$. Note that an accurate prediction requires measuring $\Phi(0)$, which can be significantly different from a standard illuminant (see Figure 3.21).

The solution given by Equation 3.80 is a generalization of Beer's law; for a purely absorbing substance when no fluorescence is present, the matrix M consists of the terms $2\ln 10 \cdot \varepsilon(\lambda_i)$ on the diagonal and of zeros anywhere else. This simplification of Equation 3.80 leads to Equation 3.23, the absorption equation for diffuse light.[43]

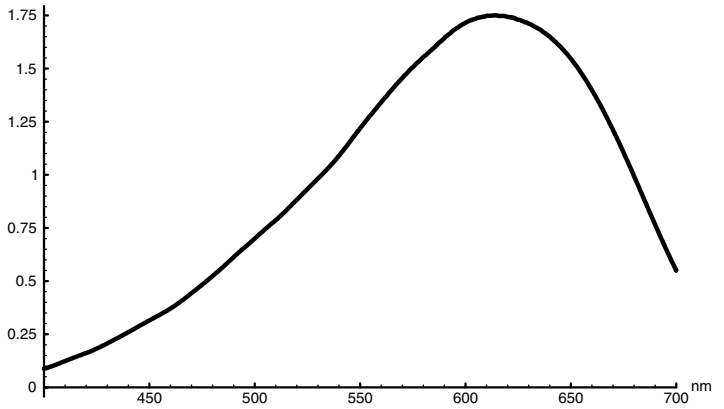


Figure 3.21 Relative radiance spectrum of the tungsten light source of a spectrophotometer. It was measured by mounting the radiometer at the position of the sample holder. Note that this spectrum is significantly different from the standard illuminant A.

This approach can be extended to cases involving two or more fluorescent substances but, for this end, we must distinguish between several possible cases. Let A and B be two different substances whose fluorescence density matrices are $\mathrm{M_A}$ and $\mathrm{M_B}$ and whose respective concentrations are c_A and c_B. Hence,

- If light goes first through a layer of thickness X_A of substance A and then through a layer of thickness X_B of substance B, we have,

$$\Phi(A, B) = \exp(-\mathrm{M_B} c_\mathrm{B} X_\mathrm{B}) \cdot \exp(-\mathrm{M_A} c_\mathrm{A} X_\mathrm{A}) \cdot \Phi(0) \tag{3.83}$$

- If light goes first through a layer of thickness X_B of substance B and then through a layer of thickness X_A of substance A, we have,

$$\Phi(B, A) = \exp(-\mathrm{M_A} c_\mathrm{A} X_\mathrm{A}) \cdot \exp(-\mathrm{M_B} c_\mathrm{B} X_\mathrm{B}) \cdot \Phi(0) \tag{3.84}$$

- If light goes through a layer of thickness X consisting of a mixture of the substances A and B, we have,

$$\Phi(\overbrace{A, B}) = \exp(-X(\mathrm{M_A} c_\mathrm{A} + \mathrm{M_B} c_\mathrm{B})) \cdot \Phi(0) \tag{3.85}$$

If either substance A or B is fluorescent, the matrices $\mathrm{M_A}$ and $\mathrm{M_B}$ do not necessarily commute, so the resulting transmittance spectrum may be different in each of these three cases. As an example, let us consider the case consisting of a yellow filter and of a fluorescent yellow filter that absorbs blue light between 400 and 500 nm and emits green light between 500 and

600 nm (see Figure 3.22). If white light goes first through the yellow filter and then through the fluorescent yellow filter (see Figure 3.22A), the blue light is absorbed by the yellow filter, and it cannot cause fluorescence in the fluorescent yellow filter. But, if white light goes first through the fluorescent yellow filter and then through the yellow filter (see Figure 3.22B), green light is produced by fluorescence in the first filter. This green light is not absorbed by the second filter, hence the resulting spectrum is different from that of the first case.

3.7.2 *From a one-flux to a two-flux model for a reflective substrate*

In the present fluorescence model,[44] we consider a fluorescent medium in optical contact with a substrate that is a diffuse reflector (see Figure 3.23). This reflector is supposed to be Lambertian.[30] Like in the Kubelka–Munk analysis, we consider an upward flux j and a downward flux i going through an infinitely thin layer of the medium, which contains a fluorescent substance at concentration c (see Figure 3.24). The positive direction of the variable x is oriented upward, and its origin is at the bottom of the fluorescent medium. To simplify the equations used in this section, let us introduce the column vector $\mathbf{j}$, whose components are the intensities of the upward flux at various wavelength, and the column vector $\mathbf{i}$, whose components are the intensities of the downward flux.

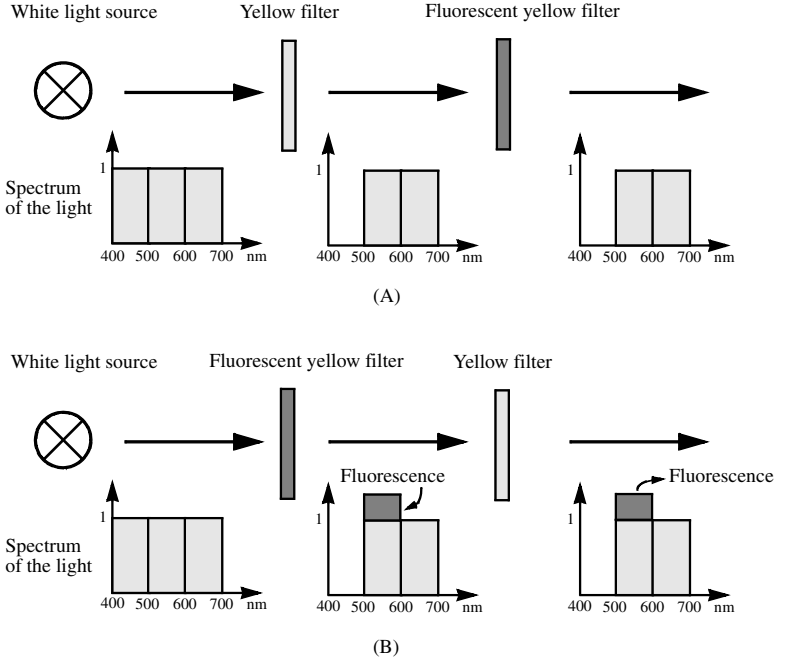


Figure 3.22 The noncommutativity of a yellow filter and a fluorescent yellow filter.

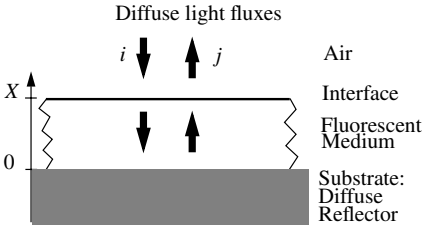


Figure 3.23 Model of a fluorescent reflector made of a fluorescent medium in optical contact with a diffuse reflector. This model describes well a high-quality paper made of an ink-absorbing layer in optical contact with a diffuse white reflector. The arrows represent diffuse light fluxes (light is coming from all directions of one hemisphere with an angular distribution corresponding to that of a Lambert surface).

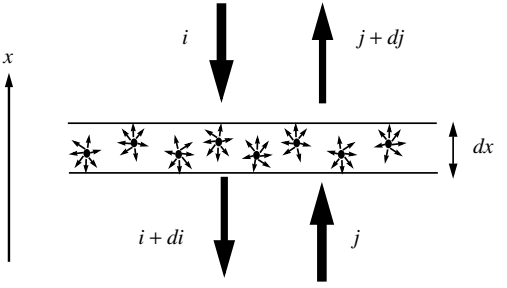


Figure 3.24 Absorption and emission in an infinitely thin layer of the transparent medium containing a fluorescent substance.

$$\mathbf{j} = \begin{bmatrix} j(\lambda_1) \\ . \\ . \\ . \\ j(\lambda_n) \end{bmatrix}, \qquad \mathbf{i} = \begin{bmatrix} i(\lambda_1) \\ . \\ . \\ . \\ i(\lambda_n) \end{bmatrix}$$

We also write the fluorescence density matrix M of the ink as the difference between a diagonal matrix A representing the absorption and a strictly lower triangular matrix F representing the fluorescent emission.

$$\mathrm{M} = \mathrm{A} - \mathrm{F}$$

$$= 2\ln 10 \left(\begin{bmatrix} \varepsilon(\lambda_1) & & 0 \\ & \ddots & \\ 0 & & \varepsilon(\lambda_n) \end{bmatrix} - \begin{bmatrix} 0 & & 0 \\ & \ddots & \\ F_{i,j} & & 0 \end{bmatrix} \right)$$

$$= 2\ln 10 \begin{bmatrix} \varepsilon(\lambda_1) & & 0 \\ & \ddots & \\ -F_{i,j} & & \varepsilon(\lambda_n) \end{bmatrix} \tag{3.86}$$

Considering first the vector of upward flux $\mathbf{j}$, its variation $(d/dx)\mathbf{j}(x)$ has two components. The first one is the absorption and fluorescent emission caused by the upward flux which is, according to the results of Section 3.7.1, $-c\mathrm{M}\mathbf{j}(x)$. The second component is the fluorescence caused by the downward flux $\mathbf{i}$, which is emitted in the upward direction, $c\mathrm{F}\mathbf{i}(x)$. Hence, the change of $\mathbf{j}$ is

$$\frac{1}{c} \cdot \frac{d}{dx}\mathbf{j}(x) = \mathrm{F}\mathbf{i}(x) - \mathrm{M}\mathbf{j}(x) \tag{3.87}$$

The same reasoning is applied to the downward flux and leads to a similar equation. Note that the downward orientation of $\mathbf{i}$ introduces a change of sign. By combining the equations obtained for $\mathbf{j}$ and $\mathbf{i}$, we get a system of differential equations whose matrix form is

$$\frac{1}{c} \cdot \frac{d}{dx} \begin{bmatrix} \mathbf{i}(x) \\ \mathbf{j}(x) \end{bmatrix} = \begin{bmatrix} \mathrm{M} & -\mathrm{F} \\ \mathrm{F} & -\mathrm{M} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{i}(x) \\ \mathbf{j}(x) \end{bmatrix} \tag{3.88}$$

Equation 3.88 is a linear differential equation of the first order with constant coefficients. When x is integrated between 0 and X, it admits a solution[23] that is given by the matrix exponential

$$\begin{bmatrix} \mathbf{i}(X) \\ \mathbf{j}(X) \end{bmatrix} = \exp\left(\begin{bmatrix} \mathrm{M} & -\mathrm{F} \\ \mathrm{F} & -\mathrm{M} \end{bmatrix} cX \right) \cdot \begin{bmatrix} \mathbf{i}(0) \\ \mathbf{j}(0) \end{bmatrix} \tag{3.89}$$

where $\mathbf{i}(X)$ and $\mathbf{j}(X)$ are, respectively, the spectra of the downward and of the upward flux at vertical location X. The matrix exponential is defined as follows:

$$\exp\left(\begin{bmatrix} \mathrm{M} & -\mathrm{F} \\ \mathrm{F} & -\mathrm{M} \end{bmatrix} cX \right) = \sum_{i=0}^{\infty} \frac{\left(\begin{bmatrix} \mathrm{M} & -\mathrm{F} \\ \mathrm{F} & -\mathrm{M} \end{bmatrix} cX \right)^i}{i!} \tag{3.90}$$

At the bottom of the fluorescent medium, the spectrum of the upward flux $\mathbf{j}(0)$ is linked with the spectrum of the downward flux $\mathbf{i}(0)$ by the relation

$$\mathbf{j}(0) = R_g \cdot \mathbf{i}(0) \tag{3.91}$$

where R_g is the reflection matrix of the substrate.[45] For pure reflectors, this matrix is diagonal, and the coefficients on the diagonal are the body reflectances of the different wavelength bands. If the reflecting substrate contains fluorescent substances (as, for instance, optical brighteners), the matrix R_g is triangular. Note that commercial bispectral spectrofluorimeters can be used to measure the matrix R_g (see Section 3.7.4).

3.7.3 Spectral prediction for reflective fluorescent material

The reflective fluorescent material made of a diffusely reflecting substrate with a fluorescent coating is modeled by means of three matrices: the Saunderson correction matrix (Equation 3.66); the matrix exponential (Equation 3.89), which models the fluorescent medium; and the reflection matrix R_g of the substrate (Equation 3.91).

By multiplying the Saunderson correction matrix with the matrix exponential, we obtain the following relation:

$$\begin{bmatrix}\mathbf{i}\\ \mathbf{j}\end{bmatrix} = \begin{bmatrix}\frac{1}{1-r_s}\mathrm{I} & \frac{-r_i}{1-r_s}\mathrm{I}\\ \frac{r_s}{1-r_s}\mathrm{I} & \left(1-r_i-\frac{r_i r_s}{1-r_s}\right)\mathrm{I}\end{bmatrix} \cdot \exp\left(\begin{bmatrix}\mathrm{M} & -\mathrm{F}\\ \mathrm{F} & -\mathrm{M}\end{bmatrix} cX\right) \cdot \begin{bmatrix}\mathbf{i}(0)\\ \mathbf{j}(0)\end{bmatrix} = \begin{bmatrix}\mathrm{T} & \mathrm{U}\\ \mathrm{V} & \mathrm{W}\end{bmatrix} \cdot \begin{bmatrix}\mathbf{i}(0)\\ \mathbf{j}(0)\end{bmatrix} \tag{3.92}$$

where T, U, V, W = matrices
I = the identity matrix

Thanks to Equation 3.91, it is possible to express the vector $\mathbf{j}$ as a function of $\mathbf{i}$.

$$\mathbf{j} = (\mathrm{V} + \mathrm{W} \cdot R_g) \cdot (\mathrm{T} + \mathrm{U} \cdot R_g)^{-1} \cdot \mathbf{i} \tag{3.93}$$

Because the multiplication of matrices is not commutative, the order of the terms in Equation 3.93 must be respected. This corresponds to the fact that superposed fluorescent layers do not commute, as we have already seen in Section 3.7.1.

If the fluorescent material is illuminated by a diffuse light source of spectrum $\mathbf{i}$, the spectrum of the diffuse reflected light is $\mathbf{j}$. The reflectance spectrum is computed by dividing the components of $\mathbf{j}$ by the components of $\mathbf{i}$.

$$R(\lambda) = \frac{j(\lambda)}{i(\lambda)} \tag{3.94}$$

Figure 3.25 shows the absorption spectrum, the fluorescence spectrum, and the reflectance spectrum of a fluorescent yellow ink printed on paper.

As in Section 7.1, this approach can be extended to cases involving two or more fluorescent substances. Once again, we must distinguish among several possible cases. Suppose A and B are two different substances whose

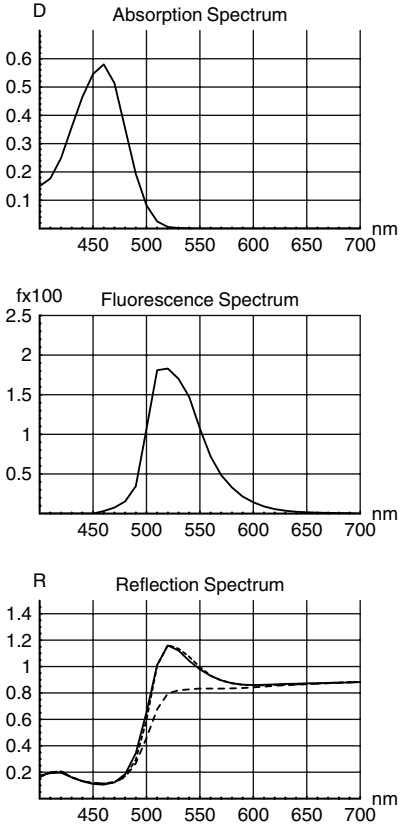


Figure 3.25 Absorption spectrum $D(\lambda) = 2\ c\ \varepsilon(\lambda)$, normalized fluorescence spectrum $f(\lambda)$, and reflection spectrum $R(\lambda)$ of a fluorescent yellow ink printed on paper. In this particular case, the excitation spectrum and the absorption spectrum are identical. The quantum yield of the yellow ink is $Q = 0.7$. The paper consists of a transparent coating of refractive index $n = 1.5$ in optical contact with a diffusely reflecting substrate without optical brighteners. The spectrum of the light source of the measuring instrument is given in Figure 3.21. The measured reflection spectrum (continuous line) is well predicted by the model (dotted line). The dashed line shows the prediction result when only absorption is taken into account.

fluorescence density matrices are $\mathrm{M_A} = \mathrm{A_A} - \mathrm{F_A}$ and $\mathrm{M_B} = \mathrm{A_B} - \mathrm{F_B}$, and whose respective concentrations are c_A and c_B. As a result,

- If a layer of thickness X_A of substance A is on top of a layer of thickness X_B of substance B, we have,

$$\begin{bmatrix}\mathbf{i}(X)\\ \mathbf{j}(X)\end{bmatrix} = \exp\left(\begin{bmatrix}\mathrm{M_A} & -\mathrm{F_A}\\ \mathrm{F_A} & -\mathrm{M_A}\end{bmatrix} c_\mathrm{A} X_\mathrm{A}\right) \cdot \exp\left(\begin{bmatrix}\mathrm{M_B} & -\mathrm{F_B}\\ \mathrm{F_B} & -\mathrm{M_B}\end{bmatrix} c_\mathrm{B} X_\mathrm{B}\right) \cdot \begin{bmatrix}\mathbf{i}(0)\\ \mathbf{j}(0)\end{bmatrix} \quad (3.95)$$

- If a layer of thickness X_B of substance B is on top of a layer of thickness X_A of substance A, we have,

$$\begin{bmatrix}\mathbf{i}(X)\\ \mathbf{j}(X)\end{bmatrix} = \exp\left(\begin{bmatrix}\mathrm{M_B} & -\mathrm{F_B}\\ \mathrm{F_B} & -\mathrm{M_B}\end{bmatrix} c_\mathrm{B} X_\mathrm{B}\right) \cdot \exp\left(\begin{bmatrix}\mathrm{M_A} & -\mathrm{F_A}\\ \mathrm{F_A} & -\mathrm{M_A}\end{bmatrix} c_\mathrm{A} X_\mathrm{A}\right) \cdot \begin{bmatrix}\mathbf{i}(0)\\ \mathbf{j}(0)\end{bmatrix} \quad (3.96)$$

- If light goes through a layer of thickness X consisting of a mixture of the substances A and B, we have,

$$\begin{bmatrix}\mathbf{i}(X)\\ \mathbf{j}(X)\end{bmatrix} = \exp\left(\left(\begin{bmatrix}\mathrm{M_A} & -\mathrm{F_A}\\ \mathrm{F_A} & -\mathrm{M_A}\end{bmatrix} c_\mathrm{A} + \begin{bmatrix}\mathrm{M_B} & -\mathrm{F_B}\\ \mathrm{F_B} & -\mathrm{M_B}\end{bmatrix} c_\mathrm{B}\right) X\right) \cdot \begin{bmatrix}\mathbf{i}(0)\\ \mathbf{j}(0)\end{bmatrix} \quad (3.97)$$

If one of the substances A or B is fluorescent, the matrices $\mathrm{M_A}$ and $\mathrm{M_B}$ do not necessarily commute, so the resulting reflectance spectrum may be different in each of these three cases. As an example, let us consider the case consisting of a yellow filter and of a fluorescent yellow filter that absorbs blue light between 400 and 500 nm and emits green light between 500 and 600 nm (see Figure 3.26). If the yellow filter is superposed on top of the fluorescent yellow filter (see Figure 3.26A), the blue light is absorbed by the yellow filter, and it cannot cause fluorescence in the resulting spectrum. But if the fluorescent yellow filter is superposed on top of the yellow filter (see Figure 3.26B), green light is produced by fluorescence in the resulting spectrum.

3.7.4 Measuring the parameters of the fluorescence model

To compute the fluorescence density matrix M, four elements have to be determined: the excitation spectrum, the absorption coefficient $\varepsilon(\lambda)$, the normalized fluorescence function $f(\lambda)$, and the quantum yield Q. (Note that M contains discrete values of the functions $\varepsilon(\lambda)$ and $f(\lambda)$.)

Because the dye concentration c is unknown, it is impossible to determine the absorption coefficient $\varepsilon(\lambda)$. However, according to Equation 3.19,

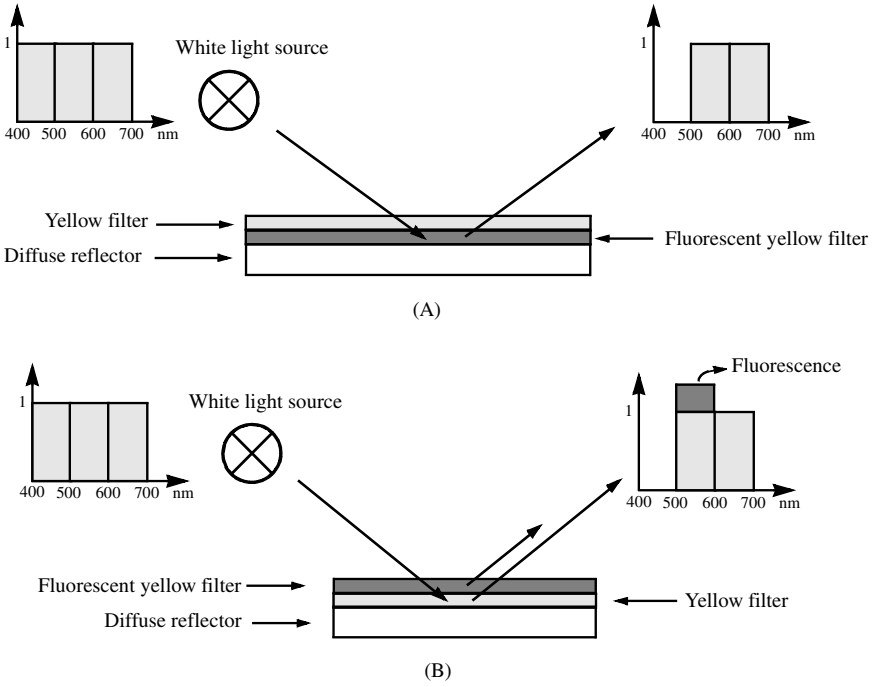


Figure 3.26 The noncommutativity of a superposition of a yellow filter and a fluorescent yellow filter covering a diffuse reflector.

the density spectrum $D(\lambda)$ and the absorption coefficient $\varepsilon(\lambda)$ are proportional and the proportionality factor is the dye surface density $q = cX$. Note that each non-zero element of the fluorescence density matrix M contains a factor $\varepsilon(\lambda_i)$ (see Equation 3.79). Because, in Equation 3.80, M is multiplied by q, each occurrence of $\varepsilon(\lambda_i)$ is multiplied by q, and this product equals the density $D(\lambda_i)$. Hence, for a given sample of density $D(\lambda_i)$, we do not need the actual values of q and of $\varepsilon(\lambda_i)$, and we can work relatively to the density spectrum $D'(\lambda_i)$ of a reference sample so that

$$D(\lambda_i) = q\varepsilon(\lambda) = q'D'(\lambda_i) \tag{3.98}$$

where q' is the proportionality factor between $D'(\lambda_i)$ and $D(\lambda_i)$.

The excitation spectrum is determined in a two-step procedure. To avoid deviations due to self-absorption, the fluorescence measurement must be performed on a sample whose maximal density is smaller than 0.1 over the whole spectrum ($D(\lambda) \leq 0.1$). This means that light emitted by fluorescence is not reabsorbed by another molecule of the sample. At first, the whole density spectrum $D(\lambda)$ of our sample is measured with a spectrophotometer. This instrument uses a monochromatic collimated light beam that goes

through the transparent sample before reaching a light detector. Because only a small fraction of the fluoresced light passes through the entrance slit of the detector, the deviation induced by the fluorescent emission can be neglected.

In the second step, the location of the excitation spectrum within the density spectrum is determined. This can be done once we have an *a priori* knowledge of the approximate position of the fluorescence spectrum (for example, by a preliminary measurement using a fluorescence spectrometer).[41] This device has two monochromators; the first one is used to generate a monochromatic light beam that excites the sample, and the second monochromator is used to analyze the light emitted by the sample. In our present measurement, the second monochromator is set to a fixed wavelength that is supposed to be within the fluorescence spectrum (the *a priori* knowledge). The first monochromator sweeps the whole spectrum, and the intensity of the emitted light is recorded. This provides the excitation spectrum and its location.

To determine the normalized fluorescence function, a fluorescence spectrometer is needed. The sample is excited with a monochromatic light beam whose wavelength corresponds to the maximum absorption in the excitation spectrum. Because the shape of the fluorescence emission spectrum does not depend on the excitation wavelength (see Section 3.7), the normalized fluorescence function is easy to compute by dividing the measured fluorescence spectrum by its integral value, which is proportional to the number of fluoresced photons.

Once the excitation spectrum and the fluorescence function have been measured, the quantum yield is determined using a method described in the literature.[46] This method is based on a measurement made relatively to a standard fluorescent substance of known quantum yield. To be reliable, the location of the excitation spectrum and the location of the fluorescence spectrum of the standard substance must correspond to those of our sample. Based on these criteria, the standard substance is chosen from tables given in the literature.[47]

The quantum yield of the unknown substance is given by[46]

$$Q_u = \frac{A_s F_u n^2}{A_u F_s n_0^2} \cdot Q_s \tag{3.99}$$

In this equation, the subscript u stands for *unknown* and the subscript s for *standard*. A is the absorption at the excitation wavelength, and Q is the quantum yield. The refractive indices of the solvent of the standard fluorescent substance (n_0) and of the medium of the unknown fluorescent substance (n) are also taken into account. The variable F is proportional to the total number of photons emitted by fluorescence. This value is computed by integrating the spectrum emitted by fluorescence during the experiment.

Within the excitation spectrum, we must select a single wavelength that gives the highest possible fluorescence in both the standard substance and our sample. These two substances are excited using the same fluorescence spectrometer at the selected wavelength, and the spectrum of the fluoresced light is measured. By integrating the fluorescence spectra of the standard substance and of our unknown sample, we get the respective number of photons, F_s and F_u, emitted by fluorescence. Because the excitation spectra of both substances are known, we have the respective absorption factors A_s and A_u at the selected excitation wavelength. Finally, the quantum yield Q_u of our sample is calculated using Equation 3.99. Note that three values must be found in the literature: the quantum yield Q_s of the standard substance,[46] the refraction index n_0 of the medium containing it, and the refraction index n of our sample's medium.

This experimental determination of the quantum yield is rather difficult to perform. Therefore, it is often preferred to estimate the quantum yield by using a best-fit method applied on a test sample. This is an iterative process. First we give Q a start value, then we compute the reflectance spectrum of the test sample and compare the result with the measured spectrum. If the fluorescence is underestimated, we increase Q; otherwise, we decrease it. The computation is then redone with the new value of Q. This iterative process stops when the square of the difference between the computed and the measured spectra is minimal. Note that this estimation method reduces the number of experiments to be performed, but it no longer guarantees that the real physical quantum yield is used.

Because the refractive index n of the fluorescent medium is known from the literature,[48] the internal and external reflection r_i and r_s can be computed using Judd's method (see Equations 3.13 and 3.14 in Section 3.3.2). The reflection matrix of the substrate $\mathrm{R_g}$ is measured using the two monochromator method described by Donaldson,[45] using barium sulfate ($\mathrm{Ba\,SO_4}$) as the white reference. Note that this measurement must be performed on an identical sample without the fluorescent coating. Sometimes, only the coated substrate without fluorescent substances in the coating is available, e.g., in the case of coated paper. In this particular case, the measured reflection matrix R corresponds to the matrix product.

$$\mathrm{R} = \left(\frac{r_s}{1-r_s}\mathrm{I} + \left(1 - r_i - \frac{r_i r_s}{1-r_s}\right)\cdot \mathrm{R}_g\right)\cdot\left(\frac{1}{1-r_s}\mathrm{I} - \frac{r_i}{1-r_s}\cdot \mathrm{R}_g\right)^{-1} \tag{3.100}$$

This relation can be solved for the matrix $\mathrm{R_g}$ as follows:

$$\mathrm{R_g} = [(1 - r_i - r_s)\mathrm{I} + r_i(1 - r_s)\mathrm{R}]^{-1}\cdot(\mathrm{R} - r_s\mathrm{I}) \tag{3.101}$$

Note that the multiplication of matrices is not commutative, so the order of the terms in Equation 3.101 must be respected.

3.8 Models for halftoned samples

Most printing devices are only *bilevel*, meaning that they are capable of printing ink only at a certain fixed density or leaving the substrate unprinted, but they cannot produce intermediate ink densities. In such devices, the visual impression of intermediate tone levels is usually obtained by means of the halftoning technique, i.e., by breaking the original continuous-tone image into small dots whose area coverage varies depending on the tone level. Halftoning is also used for most color printing devices, where each of the inks [usually cyan (C), magenta (M), yellow (Y), and often black (K)] is only bilevel. This gives to the eye, when looking from a sufficient distance, an illusion of a full range of intermediate color levels, although the printing device is only bilevel. In this section, we focus our discussion on predicting the reflectance of halftoned samples, where dyes (or pigments) are no longer uniformly distributed over the entire surface.

3.8.1 The Murray–Davis equation

Let us consider a surface of unit area, and let $R_s(\lambda)$ be the reflectance spectrum of a *solid* sample, i.e., a sample whose surface is fully covered with an ink layer of constant density. The reflectance spectrum of the bare substrate is denoted $R_g(\lambda)$. The total reflectance spectrum $R(\lambda)$ of a halftoned sample having a fraction a of area covered with ink $(0 \le a \le 1)$ is given by the following weighted sum (see Figure 3.27):

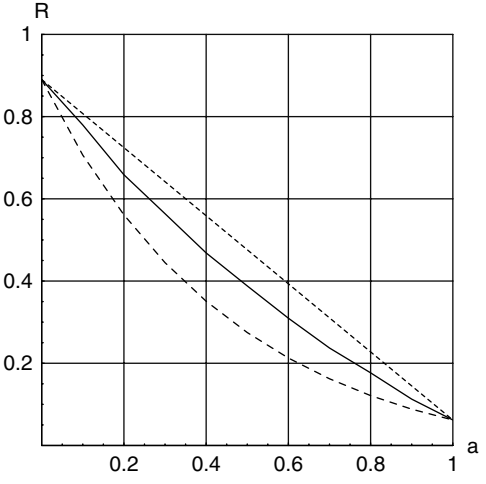


Figure 3.27 Reflectance $R(\lambda = 550\ \text{nm})$ of halftoned samples having a fraction a of area covered with black ink (continuous line). The Murray–Davis model assumes a linear behavior (dotted line), whereas the Clapper–Yule equation predicts a nonlinear behavior caused by the light scattering in the substrate (dashed line).

$$R(\lambda) = (1-a)R_g(\lambda) + aR_s(\lambda) \tag{3.102}$$

This relation is often written in a different way using the *reflection density spectrum* $D(\lambda) = -\log_{10}R(\lambda)$. In this case, Equation 3.102 is called the Murray–Davis equation,[49]

$$D(\lambda) = -\log_{10}[(1-a)10^{-D_g(\lambda)} + a10^{-D_s(\lambda)}] \tag{3.103}$$

This equation is sometimes given in an alternative form that is useful for converting densities into area coverage.

$$a = \frac{1-10^{-D(\lambda)-D_g(\lambda)}}{1-10^{-D_s(\lambda)-D_g(\lambda)}} \tag{3.104}$$

3.8.2 The classical Neugebauer theory

In 1937, Neugebauer proposed a method for predicting the spectra of halftoned color prints produced by the superposition of cyan, magenta, and yellow *dot-screens*.[50] In traditional printing, a dot-screen is a regular lattice of dots that are ordered in parallel rows along two perpendicular axes. The dots have variable sizes so as to produce the correct halftone levels. Furthermore, the cyan, magenta, and yellow screens are mutually rotated by 30° or 60° to avoid moiré patterns.[51] Neugebauer observed, under the microscope, that such a halftone print was in fact a mosaic of eight colors, which correspond to the 2^3 possible overlaps of the cyan, magenta, and yellow inks: white (= no ink), cyan, magenta, yellow, red, green, blue, and black (see Table 3.2). These colors are called *Neugebauer primaries*. Neugebauer based his model on the assumption that the dots in the different screens are almost independent of each other. This assumption, attributed to Demichel,[52] is, however, only approximately true in traditional color printing.[53]

To explain Neugebauer's method, let c, m, and y be the fractions of area covered by the cyan ink, the magenta ink, and the yellow ink, respectively. From a statistical point of view, c, m, and y can also be interpreted as the probabilities for a given point to be covered by one of the three inks. Hence, the probability for a given point to be white, i.e., not covered by any ink, equals $1-c$ (no cyan ink) times $1-m$ (no magenta ink) times $1-y$ (no yellow ink). By a similar reasoning, we deduce the fraction of area occupied by the eight Neugebauer primaries as shown in Table 3.2. The reflectance spectrum of the halftoned color print is then given by the following Neugebauer equation:

$$R(\lambda) = \sum_{j=1}^{8} a_j R_j(\lambda) \tag{3.105}$$

Table 3.2 Fraction of Area Occupied by the Eight Primaries of the Neugebauer Model

Primary	Ink Combination	Reflectance	Fraction of Area
White	—	$R_1(\lambda)$	$a_1 = (1-c)(1-m)(1-y)$
Cyan	Cyan	$R_2(\lambda)$	$a_2 = c(1-m)(1-y)$
Magenta	Magenta	$R_3(\lambda)$	$a_3 = (1-c)m(1-y)$
Yellow	Yellow	$R_4(\lambda)$	$a_4 = (1-c)(1-m)y$
Red	Magenta, yellow	$R_5(\lambda)$	$a_5 = (1-c)my$
Green	Cyan, yellow	$R_6(\lambda)$	$a_6 = c(1-m)y$
Blue	Cyan, magenta	$R_7(\lambda)$	$a_7 = cm(1-y)$
Black	Cyan, magenta, yellow	$R_8(\lambda)$	$a_8 = cmy$

This is a simple extension of Equation 3.102. Note that Equation 3.105 is a polynomial of degree three for the dot area triplet (c, m, y).

The classical Neugebauer equation leads to color prediction errors of about $\Delta E = 10$ in CIELAB. Several attempts have been made to improve the Neugebauer model.[54] One of the most important improvements is the cellular Neugebauer method proposed in 1992 by Heuberger et al.[55] The CMY color space is subdivided into rectangular cells. The reflectance spectra $R(\lambda)$ of the samples corresponding to the corners of the cells are measured. By means of Equation 3.105, the new reflectance spectra $\tilde{R}_j(\lambda)$ $(1 \le j \le 8)$ of eight equivalent primaries according to Table 3.2 are computed for each cell. The reflectance spectrum $R'(\lambda)$ of a new given color defined by the dot area triplet (c', m', y') is now computed in a two-step process. First, we find the cell of the CMY color space to which it belongs; then, we assign the dot area triplet (c', m', y') into the traditional Neugebauer equation with the equivalent reflectances $\tilde{R}_j(\lambda)$ that we have computed for the cell.[56] Note that this corresponds to a polynomial interpolation of degree three within each cell. Using this improved model, the average prediction error drops to $\Delta E = 3$ in CIELAB when the color space is subdivided into $4^3 = 64$ cells; however, this requires measuring $5^3 = 125$ samples. The main drawback of this cellular method lies, indeed, in the large number of samples that must be measured.

3.8.3 *Extended Neugebauer theory*

In some printing processes, the number of inks k is greater than three, and each ink may have m density levels $(m \ge 2)$. The Neugebauer theory can be easily generalized to such cases by considering each of the m^k possible ink

superpositions as a Neugebauer primary. The generalized Neugebauer equation thus obtained is:

$$R(\lambda) = \sum_{j=1}^{m^k} a_j R_j(\lambda) \text{ with } \sum_j a_j = 1 \tag{3.106}$$

where $R_j(\lambda)$ = reflectance of the Neugebauer primary j

a_j = fraction of area it occupies

Note that, if the superposed layers are not independent of each other, the parameters a_j cannot be calculated as in the classical Neugebauer model.

3.8.4 The Yule–Nielsen equation

Yule and Nielsen pointed out that light does not emerge from the substrate at the point where it entered. This is a consequence of the light scattering in the substrate. Therefore, a photon that penetrates the substrate in an area without ink may emerge in an inked area, and vice versa. As a consequence of this exchange of photons, the fraction of area a obtained from Equation 3.104 (the Murray–Davis equation) is greater than the real area covered by ink. This phenomenon is called *optical dot gain* or the *Yule–Nielsen effect*.

To improve the prediction of the reflection density $D(\lambda)$ of a halftoned print, in 1951, Yule and Nielsen suggested the following correction to Equation 3.103:

$$D(\lambda) = -n\log_{10}\left[(1-a)10^{-\frac{D_g(\lambda)}{n}} + a10^{-\frac{D_s(\lambda)}{n}}\right] \tag{3.107}$$

where $D_g(\lambda)$ = reflectance density of the substrate

a = fraction of area covered by the ink whose solid reflectance density is $D_s(\lambda)$

n = an empirical correction factor called the *Yule–Nielsen factor*

Factor n must be determined experimentally and depends on the optical properties of the substrate. In the literature, Equation 3.107 is called the *Yule–Nielsen equation*.[57,58] Note that, in the particular case of $n = 1$, Equation 3.107 gives the Murray–Davis equation (Equation 3.103). The generalization of Equation 3.107 for m^k Neugebauer primaries was suggested by Viggiano.[59] In the literature, this generalization is called the *n-modified Neugebauer equation*, and it can be written as follows:

$$[R(\lambda)]^{\frac{1}{n}} = \sum_{j=1}^{m^k} a_j [R_j(\lambda)]^{\frac{1}{n}} \tag{3.108}$$

3.8.5 *The Clapper–Yule equation*

After formulating the Yule–Nielsen equation, Yule worked with Clapper to develop an accurate model for halftone prints, based on a theoretical analysis taking into account surface-reflection, multiple-scattering, internal-reflection, and ink transmission.[60] In this model, light is reflected many times internally by the air–medium interface and by the substrate. A fraction of light emerges at each reflection cycle, and the total reflectance is the sum of all those fractions.

Let us denote r_s as the surface reflection, r_i as the internal reflection, ρ_g as the body reflectance of the substrate, and $T(\lambda)$ as the transmittance spectrum of the ink under diffuse light. A light beam that strikes the surface of a halftone print is partially reflected and partially transmitted into the medium (see Figure 3.28). The reflected fraction is given by the surface reflection r_s. The transmitted fraction has two components. The first component $(1-r_s)(1-a)$ enters the medium through the unprinted area; the second component $(1-r_s)aT(\lambda)$ enters the medium through the ink of transmittance $T(\lambda)$. Therefore, the irradiance in the substrate resulting from the entering light is $T_0 = (1-r_s)(1-a+aT(\lambda))$. The light is assumed to be totally scattered within the substrate of body reflectance ρ_g. The emerging light emerging is again attenuated by a factor $(1-a+aT(\lambda))$ as a result of the ink pattern, and by a factor $(1-r_i)$ because of the internal reflection. The first emergence of light is given by $(1-r_s)(1-r_i)[1-a+aT(\lambda)]^2\rho_g$. The

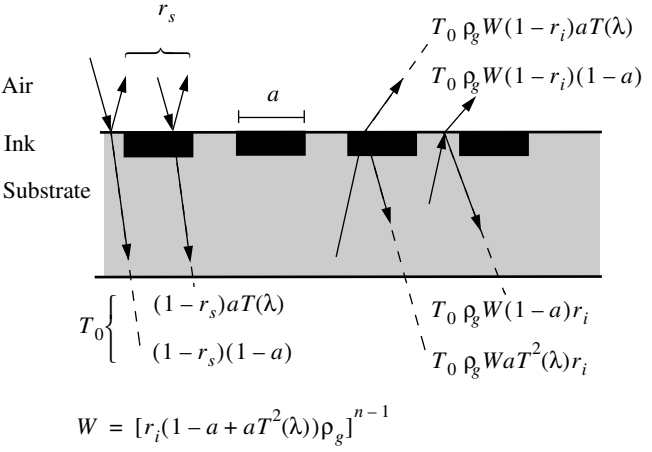


Figure 3.28 In the Clapper–Yule model, fractions of light emerge at each reflection cycle.

internally reflected light suffers one further change. The fraction that attempted to emerge through an inked area must pass through the ink a second time, so its intensity must be multiplied by $T(\lambda)$. The light that reenters the substrate after the internal reflection is then given by $(1-r_s)[(1-a)+aT(\lambda)]\rho_g[r_i(1-a+aT^2(\lambda))]$. This sequence of events continues until the remaining light is negligible. The emerging fractions of light are as follows (see Figure 3.28):

- Surface reflection,

$$r_s$$

- First emergence,

$$(1-r_s)(1-r_i)[1-a+aT(\lambda)]^2\rho_g$$

- Second emergence,

$$(1-r_s)(1-r_i)[1-a+aT(\lambda)]^2\rho_g[r_i(1-a+aT^2(\lambda))\rho_g]$$

- Third emergence,

$$(1-r_s)(1-r_i)[1-a+aT(\lambda)]^2\rho_g[r_i(1-a+aT^2(\lambda))\rho_g]^2$$

- $\ldots n$ th emergence,

$$(1-r_s)(1-r_i)[1-a+aT(\lambda)]^2\rho_g[r_i(1-a+aT^2(\lambda))\rho_g]^{n-1}$$

The sum of this geometrical series is the reflectance of the halftone print (see Figure 3.27). This leads to the following Clapper–Yule equation, which was published in 1953:[60]

$$R(\lambda) = r_s + \frac{\rho_g(1-r_s)(1-r_i)(1-a+aT(\lambda))^2}{1-\rho_g r_i(1-a+aT^2(\lambda))} \tag{3.109}$$

3.8.6 *Advanced models*

The Yule–Nielsen effect has a large impact on the color produced by halftone prints. Intensive investigations have been made so as to relate the empirical parameter n of the Yule–Nielsen equation to physical quantities. The resulting theories model the light scattering in the substrate by a point spread function (PSF) $P(x, y)$, which expresses the density of probability for a pho-

ton entering the substrate at location $(0, 0)$ to emerge at the location (x, y). The light reflected at location (x, y) of a halftone print is then given by:

$$\begin{aligned} R(x, y) &= r_s + (1 - r_s)T(x, y)P(x, y)^*T(x, y)\rho_g \\ &= r_s + (1 - r_s)T(x, y)\rho_g \iint P(x - x', y - y')T(x', y')dx'dy' \end{aligned} \tag{3.110}$$

where $*$ = convolution operator

ρ_g = body reflectance of the substrate

r_s = surface reflection

$T(x, y)$ = transmittance at location (x, y)

If there is ink in this location, then $T(x, y) = T(\lambda)$; otherwise, $T(x, y) = 1$. The wavelength designation (λ) is dropped to simplify the notation, but $P(x, y)$, $T(x, y)$, and $R(x, y)$ are functions of wavelength. The reflectance $R(\lambda)$ of the whole halftone print is the spatial average of $R(x, y)$. Note that the multiple internal reflections are accounted by the PSF $P(x, y)$.

In 1978, Ruckdeschel and Hauser derived the empirical Yule–Nielsen factor n from the PSF and the period of the halftone screen.[61] They assumed a Gaussian PSF,

$$P(x, y) = \frac{1}{\pi\sigma^2}\exp\left[-\left(\frac{x^2 + y^2}{\sigma^2}\right)\right] \tag{3.111}$$

where σ is a characteristic scattering length of the photon in the substrate.

According to their calculations, the Yule–Nielsen factor is given by the following relation:

$$n \approx 2 - \exp\left(-\pi\frac{\sigma}{L}\right) \tag{3.112}$$

where L = period of the screen

Note that the value of n approaches 1 as the substrate approaches a specular surface ($\sigma = 0$, Murray–Davis model; see Section 3.8.1) and approaches 2 as the substrate becomes a perfect diffuser ($\sigma = \infty$, Clapper–Yule model, see Section 3.8.5). In 1997, Rogers showed that the characteristic scattering length σ is related to two physical parameters:[62] the absorption in the substrate and the optical thickness of the substrate. If there is no absorption in the substrate, σ increases without bound as the optical thickness of the substrate tends to infinity. If absorption occurs, however, the scattering length σ reaches a limit.

Further investigations made by Rogers showed that the PSF is a series of convolutions whose terms are the contributions of the multiple internal reflections occurring in the substrate.[63]

$$P(x,y) = (1+\Gamma_i)[(p(x,y)+p(x,y))*(T^2(x,y)p(x,y))(\rho_g r_i) + p(x,y)*(T^2(x,y)p(x,y))*(T^2(x,y)p(x,y))(\rho_g r_i)^2 + \ldots] \quad (3.113)$$

where r_i = internal reflection

$p(x,y)$ = internal point spread function (internal PSF), which does not take multiple internal reflections into account

Note that, for substrates having a low internal reflection r_i, we have $P(x,y) \approx p(x,y)$. On a macroscopic scale, the PSF $P(x,y)$ shown in Equation 3.113 induces Yule–Nielsen factors n that are greater than 2. Such Yule–Nielsen factors are often found in practice, and they are not explained by the simple PSF given by Equation 3.111.

The internal PSF derives from the radiative transfer equation (see Section 3.6.1), but its analytical form, which is a series in MacDonald functions (also called modified Bessel functions), is cumbersome.[62] According to Gustavson's studies,[64] the internal PSF is closely approximated by a function $p(r)$ that has a circular symmetry ($r = \sqrt{x^2+y^2}$) and a strong radial decay.

$$p(r) = \frac{1}{2\pi d r}\exp\left[-\frac{r}{d}\right] \quad (3.114)$$

where d controls the radial extent of the internal PSF. In practice, d is computed from the light profile measured across an optically sharp edge between an inked and a non-inked area.[65,66]

In 1997, Arney proposed a probabilistic approach that is less complex than the PSF convolution.[67,68] He introduced the scattering probability $\delta_{i,j}$ for a photon that enters the substrate through a region covered by the Neugebauer primary j of transmittance T_j, to emerge through a region covered by the Neugebauer primary i of transmittance T_i. The reflectance $R(\lambda)$ of a halftone print is then given by

$$R(\lambda) = R_g \sum_i \left(a_i T_i \sum_j a_j T_j \delta_{i,j} \right) \quad (3.115)$$

where R_g = reflectance of the substrate

a_i = fraction of area covered by the Neugebauer primary i

In the particular case of traditional halftone screens, Arney showed that the scattering probabilities $\delta_{i,j}$ are given by the following empirical relations:

$$\begin{cases} \delta_{j,j} = 1-(1-a_j)[1-(1-a_j)^w+(1-a_j^w)] \\ \delta_{i,j} = (1-\delta_{j,j})\left(\dfrac{a_i}{1-a_j}\right) \end{cases} \tag{3.116}$$

where w = an empirical parameter.

Arney suggested that w is related to the characteristic scattering distance σ and the period L of the screen by $w = 1-\exp[-\alpha(\sigma/L)]$, where α is an experimentally determined proportionality coefficient. Note that the scattering probabilities $\delta_{i,j}$ can also be computed from the PSF convolution.

3.8.7 *The Monte-Carlo method*

The reflectance of a medium that is inhomogeneous or anisotropic (e.g., biological tissue) can be computed by a Monte-Carlo simulation.[69] The medium is subdivided into volume elements called *voxels*. Each voxel is associated with an absorption coefficient and a scattering phase function (see Section 3.6.1).

The computer casts a virtual ray of unit intensity on the voxels. When this ray enters a new voxel, its new intensity is computed from the absorption coefficient of the voxel, and a random number is generated to decide, according to the scattering phase function, in which direction the ray should be scattered. The process is iterated until the ray leaves the voxels or until the intensity of the ray drops below some predefined threshold. The reflectance of the medium is deduced from the results of a large number of simulations.

3.9 *New mathematical framework for color prediction of halftones*

The Kubelka–Munk model presented in the Section 3.6.2 assumes that the coating medium is uniform, i.e., that the same amount of dye is everywhere. In halftoned prints, this is no longer true, because ink is not applied uniformly over the whole surface. A photon can penetrate the printed media through an inked region and leave the printed media through a non-inked region, or vice versa (see the Yule–Nielsen effect in Section 3.8.4).

In this section, we generalize the models presented in Sections 3.6, 3.7, and 3.8, and we incorporate them into a new mathematical framework based on matrices.[70] For the sake of simplicity, we consider only two Neugebauer primaries: inked and non-inked. In case of colored samples, more primaries must be considered. Furthermore, because the ink layer is very thin (less than 10 μm), we assume that the exchange of photons between inked and non-

inked areas takes place only in the substrate. We also assume that the ink layer behaves according to the Kubelka–Munk model described previously.

Let us now consider such a surface having only two different inking levels. As in the Kubelka–Munk model, we define for each inking level two light fluxes: i_k, which is oriented downward, and j_k, which is oriented upward. The index k takes the value 0 for the non-inked region and 1 for the inked region (see Figure 3.29). Note that we drop the wavelength designation (λ) to simplify the notation, but T, ρ_g, as well as i_k, j_k, S_k, and K_k, are all functions of wavelength.

The matrix Equation 3.49 can be extended to take several inking levels into account. Let us denote M_{KM} as this extended block matrix. For two inking levels, the equation can be written as follows:

$$\frac{d}{dx}\begin{bmatrix} i_0(x) \\ j_0(x) \\ i_1(x) \\ j_1(x) \end{bmatrix} = M_{KM} \cdot \begin{bmatrix} i_0(x) \\ j_0(x) \\ i_1(x) \\ j_1(x) \end{bmatrix} = \begin{bmatrix} K_0 + S_0 & -S_0 & 0 & 0 \\ S_0 & -(K_0 + S_0) & 0 & 0 \\ 0 & 0 & K_1 + S_1 & -S_1 \\ 0 & 0 & S_1 & -(K_1 + S_1) \end{bmatrix} \cdot \begin{bmatrix} i_0(x) \\ j_0(x) \\ i_1(x) \\ j_1(x) \end{bmatrix} \quad (3.117)$$

where K_0, S_0, K_1, and S_1 are, respectively, the absorption and scattering coefficients of the non-inked medium and the inked medium. By integrating Equation 3.117 between $x = 0$ and $x = X$, we get

$$\begin{bmatrix} i_0(X) \\ j_0(X) \\ i_1(X) \\ j_1(X) \end{bmatrix} = \exp(M_{KM} \cdot X) \cdot \begin{bmatrix} i_0(0) \\ j_0(0) \\ i_1(0) \\ j_1(0) \end{bmatrix} \quad (3.118)$$

The definition of the matrix exponential is given in Equation 3.51.

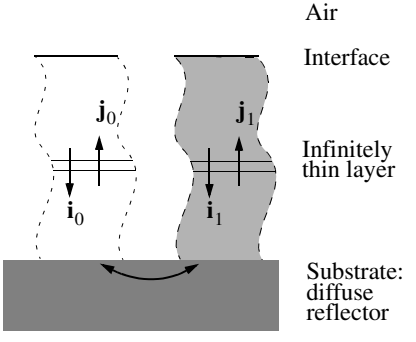


Figure 3.29 A schematic model of the printed surface. On top of the substrate, each surface element is considered to be a uniform layer which behaves according to the Kubelka–Munk model.

To take into consideration the multiple internal reflections, the Saunderson correction must also be applied here. Note that, in our case, the ink is inside the medium and not on top of it. Hence, the interface between the air and the ink-absorbing medium is the same in non-inked regions as in inked regions. Therefore, from Equation 3.66, we can directly derive the resulting Saunderson correction matrix M_{SC}.

$$\begin{bmatrix} i_0 \\ j_0 \\ i_1 \\ j_1 \end{bmatrix} = M_{SC} \cdot \begin{bmatrix} i_0(X) \\ j_0(X) \\ i_1(X) \\ j_1(X) \end{bmatrix} = \begin{bmatrix} \frac{1}{1-r_s} & \frac{-r_i}{1-r_s} & 0 & 0 \\ \frac{r_s}{1-r_s} & \left(1-r_i-\frac{r_s r_i}{1-r_s}\right) & 0 & 0 \\ 0 & 0 & \frac{1}{1-r_s} & \frac{-r_i}{1-r_s} \\ 0 & 0 & \frac{r_s}{1-r_s} & \left(1-r_i-\frac{r_s r_i}{1-r_s}\right) \end{bmatrix} \cdot \begin{bmatrix} i_0(X) \\ j_0(X) \\ i_1(X) \\ j_1(X) \end{bmatrix} \quad (3.119)$$

The key to our model lies in the way optical dot gain is expressed mathematically. Because we assume that the exchange of photons takes place only in the substrate, the optical dot gain affects only the boundary conditions at $x = 0$. This implies that the upward-oriented fluxes $j_0(0)$ and $j_1(0)$ depend on both downward-oriented fluxes $i_0(0)$, $i_1(0)$ and the body reflectance ρ_g of the substrate. This can be written in a general way in matrix form as follows:

$$\begin{bmatrix} j_0(0) \\ j_1(0) \end{bmatrix} = \rho_g \cdot \begin{bmatrix} \delta_{0,0} & \delta_{0,1} \\ \delta_{1,0} & \delta_{1,1} \end{bmatrix} \cdot \begin{bmatrix} i_0(0) \\ i_1(0) \end{bmatrix} \quad (3.120)$$

where the coefficient $\delta_{u,v}$ represents the overall probability of a photon entering through a surface element having the inking level v to emerge from a surface element having the inking level u. Note that the probability is taken throughout the full sample area. This probabilistic approach was introduced by Arney (see Section 3.8.6). Because we deal with probabilities, the sum of the coefficients $\delta_{u,v}$ belonging to the same line of the matrix in Equation 3.120 must equal 1. The computation of the scattering probabilities $\delta_{u,v}$ will be addressed in Section 3.9.2.

Now we can put all elements together and write the matrix equation of our new prediction model. By combining Equations 3.118 through 3.120 we obtain

$$\begin{bmatrix} i_0 \\ j_0 \\ i_1 \\ j_1 \end{bmatrix} = M_{SC} \cdot \exp(M_{KM} \cdot X) \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \delta_{0,0} & 0 & \delta_{0,1} \\ 0 & 0 & 1 & 0 \\ 0 & \delta_{1,0} & 0 & \delta_{1,1} \end{bmatrix} \cdot \begin{bmatrix} i_0(0) \\ \rho_g i_0(0) \\ i_1(0) \\ \rho_g i_1(0) \end{bmatrix} \quad (3.121)$$

The first matrix of Equation 3.121 represents the Saunderson correction, the second matrix corresponds to the Kubelka–Munk modeling of the ink-absorbing layer, and the third matrix models the light scattering in the substrate.

Computing the emerging fluxes j_0 and j_1 as functions of the incident fluxes i_0 and i_1 requires rearranging the lines and columns of the matrices. To keep the block structure of the matrices, we introduce a *change of basis matrix* as shown in Equation 3.122. Note that this particular change of basis matrix is its own inverse. Furthermore, the last vector of Equation 3.121 is written in Equation 3.122 as the product of a 4×2 matrix by a two-dimensional vector.

$$\begin{bmatrix} i_0 \\ i_1 \\ j_0 \\ j_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \mathrm{M}_{SC} \cdot \exp(\mathrm{M}_{KM} \cdot X)$$

$$\cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \delta_{0,0} & \delta_{0,1} \\ 0 & 0 & \delta_{1,0} & \delta_{1,1} \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ \rho_g & 0 \\ 0 & \rho_g \end{bmatrix} \cdot \begin{bmatrix} i_0(0) \\ i_1(0) \end{bmatrix} \tag{3.122}$$

After computing the matrix products in Equation 3.122, we get a 4×2 matrix that can be split into two 2×2 matrices. The first matrix relates the vector $[i_0, i_1]$ to $[i_0(0), i_1(0)]$, and the second matrix relates $[j_0, j_1]$ to $[i_0(0), i_1(0)]$. By multiplying the second matrix by the inverse of the first matrix, we derive a relation that expresses the emerging fluxes j_0 and j_1 as linear functions of the incident fluxes i_0 and i_1.

Because the incident light has the same intensity on inked and non-inked regions, we have $i_0 = i_1 = i$. Let a_1 be the inked fraction of area and $a_0 = 1 - a_1$ be the non-inked fraction of area. As in the Neugebauer model (see Equation 3.106), the reflectance spectrum $R(\lambda)$ of the whole surface is given by the weighted sum of the emerging light divided by the incident light, where the weights are the area coverages of the various primaries. Hence, the final result is given by

$$R(\lambda) = \frac{\begin{bmatrix} a_0 & a_1 \end{bmatrix} \cdot \begin{bmatrix} j_0 \\ j_1 \end{bmatrix}}{\begin{bmatrix} a_0 & a_1 \end{bmatrix} \cdot \begin{bmatrix} i_0 \\ i_1 \end{bmatrix}} = \frac{(1 - a_1) j_0 + a_1 j_1}{i} \tag{3.123}$$

3.9.1 Some particular cases of interest

Let us consider the particular case in which the average lateral light scattering distance is large compared to the size of the halftoning element. This is the assumption of complete scattering. In this case, for any inking level v, the probability $\delta_{u,v}$ equals the fraction of area a_u occupied by the inking level u,

$$\delta_{0,0} = \delta_{1,0} = a_0 = 1 - a_1 \text{ and } \delta_{0,1} = \delta_{1,1} = a_1 \tag{3.124}$$

By introducing the relations of Equation 3.124 in Equation 3.122 and assuming that $S_0 = 0$, $S_1 = 0$, $K_0 = 0$, we obtain from Equation 3.123 the well-known Clapper–Yule relation (see Equation 3.109),

$$R(\lambda) = r_s + \frac{\rho_g(1-r_s)(1-r_i)(1-a_1+a_1T(\lambda))^2}{1-\rho_g r_i(1-a_1+a_1T^2(\lambda))} \tag{3.125}$$

where $T = \exp[-K_1X]$. Note that this derivation requires the help of a mathematics software package.

In another particular case, lateral light scattering can be neglected. Hence, the probability of a photon being scattered in a region with a different inking level equals 0. This implies that $\delta_{u,u} = 1$ and $\delta_{u,v} = 0$ for $u \neq v$. In other words, the second to last matrix of Equation 3.122 is an identity matrix. In this case, assuming $S_0 = 0$, $S_1 = 0$, $K_0 = 0$, $r_i = 0$, $r_s = 0$ leads to the Murray–Davis relation (see Equation 3.102),

$$R(\lambda) = R_g[(1-a_1)+a_1T^2] \tag{3.126}$$

where $T = \exp[-K_1X]$. Note that $R_g = \rho_g$ because $r_i = 0$ and $r_s = 0$.

In the case of a fluorescent ink or of a fluorescent substrate, each element of the matrices in Equation 3.122 must be replaced by a matrix. Let us denote $\mathbf{M}$ as the fluorescence density matrix of the fluorescent ink as defined in Equation 3.86, and $\mathbf{R}_g$ as the reflection matrix. Furthermore, let us denote $\mathbf{I}$ as the identity matrix, which has the same dimension as the fluorescence density matrix $\mathbf{M}$ and the reflection matrix $\mathbf{R}_g$.

In the Kubelka–Munk matrix $\mathbf{M}_{KM}$, the sum $K_1 + S_1$ has to be replaced by M, and S_1 has to be replaced by the matrix $\mathbf{F}$ defined in Equation 3.86. In the same way, $K_0 + S_0$ and S_0 must be replaced by matrices. If the ink-absorbing medium is nonfluorescent, the scalar values $K_0 + S_0$ and S_0 are multiplied by the identity matrix $\mathbf{I}$. The body reflectance ρ_g is replaced by the reflection matrix $\mathbf{R}_g$ introduced in Section 3.7.2. In the Saunderson correction matrix $\mathbf{M}_{SC}$, each element is replaced by its scalar value multiplied by $\mathbf{I}$. The same kind of substitution must be done in the change of basis

matrix, its inverse matrix, and the matrix containing the scattering probabilities $\delta_{u,v}$. Note that i, j, $i_0(0)$, and $i_1(0)$ are replaced by vectors whose number of components equals the number of columns of the matrix M. Hence, the vector $\mathbf{i}_0 = \mathbf{i}_1 = \mathbf{i}$ equals the spectrum of the incident light source, and $(1 - a_1)\mathbf{j}_0 + a_1\mathbf{j}_1$ equals the spectrum of the reflected light.

3.9.2 *Computing the area fractions and the scattering probabilities*

The area fractions a_u and the scattering probabilities $\delta_{u,v}$ are computed by a numerical simulation.[70] High-resolution grids model the printed surface, one grid being used for each ink. The value of a grid point corresponds to the local amount of a given dye (see Figure 3.30C). In the particular case of inkjet printing, the density profile of an isolated ink impact, which was measured under the microscope, can be approximated by a parabolic function.[71] A single dot is modeled as a stamp (see Figure 3.30B).

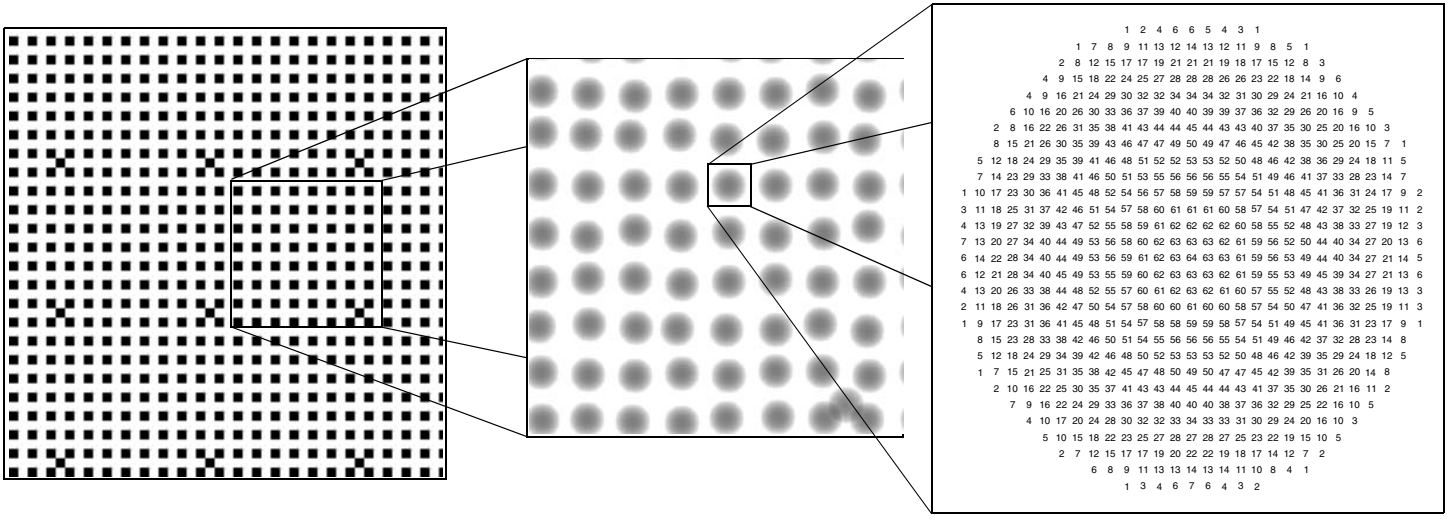
Digital printing systems use halftoning or dithering algorithms to determine whether a given location on the printed surface must be covered by a dot. To simulate accurately a given printing system, the same halftoning or dithering algorithm must be used to provide the locations of all printed dots (Figure 3.30A). Wherever a dot is printed, the model is stamped at the corresponding location on the high-resolution grid. In the particular case of inkjet printing, stamp overlapping is additive.

In a color print using k inks, k halftoned ink layers are used. The ink combination covering a surface element at position (x, y) is given by the set of k values of the grid points (x, y) in the k superposed high-resolution grids. The area covered by a given combination of k inks is estimated by counting the number of grid points having the same set of k values. The fraction of area a_u is determined by counting the number of grid points that belong to the same inking level u.

The light-scattering process can be seen as an exchange of photons between a grid point and its neighbors. As we saw in Section 3.8.6, it can be modeled by an internal point spread function $p(x, y)$ that expresses the density of probability for a photon entering at location $(0, 0)$ to emerge at location (x, y). The discrete form of the internal PSF gives the probability for an entering photon to emerge from another grid point. The function suggested by Gustavson[64] is a good approximation of the internal PSF (see Equation 3.114). The scattering probability $\delta_{u,v}$ equals the weighted sum over the whole grid of points having the inking level u with a neighbor having an inking level v. The weights of the neighbors are given by our discrete internal PSF.

3.10 *Concluding remarks*

By using a global approach, all classical color prediction models were unified within a mathematical framework based on matrices. This matrix framework provides a new insight into color prediction by modeling a reflective surface



(A) Locations of printed ink drops

(B) Printed surface simulated on a high resolution grid

(C) Stamp of a single dot

Figure 3.30 A high-resolution grid used for modeling the printed surface. The value of a grid point corresponds to the local amount of dye.

by three matrices: the Saunderson correction matrix, the Kubelka–Munk matrix, and the light-scattering matrix. This approach also allows us to predict colors with a higher accuracy because a larger number of physical phenomena are taken into account.

However, this generalized approach does not provide the ultimate answer to all color prediction needs. Several important physical phenomena are not yet taken into account; and, effects induced by the surface roughness, metallic pigments, or pearlescent pigments cannot be predicted. Nevertheless, the unified framework is more powerful than a collection of separate classical models put side by side, because it also provides solutions for difficult cases such as, for example, halftones printed with fluorescent inks or printing with a large number of nonstandard inks. In most cases, solutions are found by considering larger matrices. The difficulty is simply turned into more work for the computer.

References

1. Wyszecki, G. and Stiles, W. S., *Color Science: Concepts and Methods, Quantitative Data and Formulae*, 2nd ed., John Wiley & Sons, New York, 1982, 2.
2. Wyszecki, G. and Stiles, W. S., *Color Science: Concepts and Methods, Quantitative* Data *and Formulae*, 2nd ed., John Wiley & Sons, New York, 1982, 274.
3. Wyszecki, G. and Stiles, W. S., *Color Science: Concepts and Methods, Quantitative Data and Formulae*, 2nd ed., John Wiley & Sons, New York, 1982, 273.
4. Born, M. and Wolf, E., *Principles of Optics*, 7th ed., Cambridge University Press, Cambridge, U.K., 1999, 195.
5. Kortüm, G., *Reflectance Spectroscopy*, Springer, Berlin, 1969, 108.
6. Nassau, K., *The Physics and Chemistry of Color*, John Wiley & Sons, New York 1983, 355.
7. Hecht, E., *Optics*, 3rd ed., Addison-Wesley, New York, 1998, 94.
8. Born, M. and Wolf, E., *Principles of Optics*, 7th ed., Cambridge University Press, Cambridge, U.K., 1999, 746
9. Born, M. and Wolf, E. *Principles of Optics*, 7th ed., Cambridge University Press, Cambridge, U.K., 1999, 42.
10. Born, M. and Wolf, E. *Principles of Optics*, 7th ed., Cambridge University Press, Cambridge, U.K., 1999, 741.
11. Judd, D. B., Fresnel reflection of diffusely incident light, *J. Natl. Bureau Standards*, 29, 329–332, 1942.
12. Nassau, K., *The Physics and Chemistry of Color*, John Wiley & Sons, New York, 1983, Chap. 8.
13. Perkampus, H.-H., *Encyclopedia of Spectroscopy*, Wiley-VCH, Weinheim, Germany, 1995, 63–64.
14. Hiemenz, P.C. and Rajagopalan, R., *Principles of Colloid and Surface Chemistry*, 3rd ed., Marcel Dekker, New York, 1997, 201–203.
15. Kortüm, G., *Reflectance Spectroscopy*, Springer, Berlin, 1969, 80.
16. Mie, G., Beiträge zur Optik trüber medien, speziell kolloidaler Metallösungen, *Annalen der Physik*, vierte Folge, Band 25, 377–445, 1908 (in German).
17. Bohren, F. C. and Huffman, D. R., *Absorption and Scattering of Light by Small Particles*, John Wiley & Sons, New York, 1983, Chap. 4.

18. Bronshtein, I. N. and Semendyayev, K. A., *Handbook of Mathematics*, Springer, Berlin, 1997, 410–416.
19. Hiemenz, P. C. and Rajagopalan, R., *Principles of Colloid and Surface Chemistry*, 3rd ed., Marcel Dekker, New York, 1997, 233.
20. Kortüm, G., *Reflectance Spectroscopy*, Springer, Berlin, 1969, 94–100.
21. Chandrasekhar, S., *Radiative Transfer*, Dover, New York, 1960.
22. Kubelka, P. and Munk, F., Ein Beitrag zur Optik der Farbanstriche, *Zeitschrift für technische Physik*, 12, 593–601, 1931 (in German).
23. Boyce, W. E. and DiPrima, R. C., *Elementary Differential Equations and Boundary Value Problems*, 6th ed., John Wiley & Sons, New York, 1997, 401–405.
24. Emmel, P., Modèles de Prédiction Couleur Appliqués à l'Impression Jet d'Encre, Ph.D. thesis No. 1857, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, 1998, 55–58, http://diwww.epfl.ch/w3lsp/publications/colour/thesis-emmel.html (in French).
25. Emmel, P. and Hersch, R. D., Towards a color prediction model for printed patches, *IEEE Computer Graphics Appl.*, 19(4), 54–60, 1999.
26. Judd, D. B. and Wyszecki, G., *Color in Business, Science and Industry*, 3rd ed., John Wiley & Sons, New York, 1975, 426–431.
27. Nobbs, J. H., in *Colour Physics for Industry*, 2nd ed., R. McDonald, The Society of Dyers and Colourists, Bradford, U.K., 1997, 306.
28. Horn, R. A. and Johnson, C. R., *Matrix Analysis*, Cambridge University Press, Cambridge, U.K., 1996, 35.
29. Horn, R. A. and Johnson, C. R., *Matrix Analysis*, Cambridge University Press, Cambridge, U.K., 1996, 38.
30. Nobbs, J. H., in *Colour Physics for Industry*, 2nd ed., R. McDonald, The Society of Dyers and Colourists, Bradford, U.K., 1997, 299.
31. Judd, D.B. and Wyszecki, G., *Color in Business, Science and Industry*, 3rd ed., John Wiley & Sons, New York, 1975, 415–417.
32. Nobbs, J. H., in *Colour Physics for Industry*, 2nd ed., R. McDonald, The Society of Dyers and Colourists, Bradford, U.K., 1997, 304–307.
33. Saunderson, J. L., Calculation of the color pigmented plastics, *J. Optical Soc. Am.*, 32, 727–736, 1942.
34. Emmel P., Modèles de Prédiction Couleur Appliqués à l'Impression Jet d'Encre, Ph.D. thesis No. 1857, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, 1998, 77–78, http://diwww.epfl.ch/w3lsp/publications/colour/thesis-emmel.html (in French).
35. Williams, F. C. and Clapper, F. R., Multiple internal reflections in photographic color prints, *J. Optical Soc. Am.*, 43(7), 595–599, 1953.
36. Mudget, P. S. and Richards, L. W., Multiple scattering calculations for technology, *Appl. Optics*, 10(7), 1485–1502, 1971.
37. Völz, H. G., *Industrial Color Testing*, Wiley-VCH, New York, 1995, Sec. 3.2.
38. Skoog, D. A., West, D. M., and Holler, F. J., *Fundamentals of Analytical Chemistry*, 6th ed., Saunders, New York, 1992, Chap. 23.
39. Perkampus, H. H., *Encyclopedia of Spectroscopy*, Wiley-VCH, Weinheim, Germany, 1995, 204.
40. Vollhardt, K. P. C. and Schore, N. E., *Organic Chemistry*, 2nd ed., W. H. Freeman, New York, 1994, Chap. 15, 549–593.
41. Perkampus, H.-H., *Encyclopedia of Spectroscopy*, Wiley-VCH, Weinheim, Germany, 1995, 202.

42. Emmel, P. and Hersch, R. D., A "one channel" spectral colour prediction model for transparent fluorescent inks on a transparent support, in *Proc. IS&T/SID 5th Color Imaging Conference*, November 17–20, 1997, Scottsdale, 70–77.
43. Kortüm, G., *Reflectance Spectroscopy*, Springer-Verlag, Berlin, 1969, 106–109.
44. Emmel, P. and Hersch, R. D., Spectral colour prediction model for a transparent fluorescent ink on paper, in *Proc. IS&T/SID 6th Color Imaging Conference*, November 17–20, 1998, Scottsdale, 116–122.
45. Donaldson, R., Spectrometry of fluorescent pigments, *Br. J. Appl. Phys.*, 5, 210–214, 1954.
46. Scaiano, J. C., *CRC Handbook of Organic Photochemistry*, Vol. I, CRC Press, Boca Raton, FL, 1989, 233–236.
47. Olmsted, J., Calorimetric determinations of absolute fluorescence quantum yields, *J. Phys. Chem.*, 83(20), 2581–2584, 1979.
48. Brandrup, J. and Immergut, E. H., Eds., *Polymer Handbook*, 3rd ed., John Wiley & Sons, New York, 1989, VI/451–VI/461.
49. Yule, J. A. C., *Principles of Color Reproduction*, John Wiley & Sons, New York, 1967, 212.
50. Neugebauer, H. E. J., Die theoretischen Grundlagen des Mehrfarbenbuchdrucks, *Zeitschrift für wissenschaftliche Photographie*, 36(4), 73–89, 1937 (in German).
51. Amidror, I., *The Theory of the Moiré Phenomenon*, Kluwer Academic Publishers, Dortrecht, The Netherlands, 2000, 61.
52. Demichel, E., *Le Procédé*, 26(3), 17–21, 1924.
53. Amidror, I. and Hersch, R. D., Neugebauer and Demichel: dependence and independence in *n*-screen superpositions for colour printing, *Color Res. Appl.*, 25(4), 267–277, 2000.
54. Rolleston, R. and Balasubramanian, R., Accuracy of various types of Neugebauer models, *Proc. IS&T/SID Color Imaging Conference: Transforms & Transportability of Color*, November 1993, Scottsdale, 32–37.
55. Heuberger, K. J., Jing, Z. M., and Persiev, S., Color transformations and lookup tables, *Proc. TAGA/ISCC*, 2, 863–881, 1992.
56. Heuberger, K. J., private communication.
57. Yule, J. A. C., *Principles of Color Reproduction*, John Wiley & Sons, New York, 1967, 215.
58. Yule, J. A. C. and Field, G. G., *Principles of Color Reproduction*, GAFT Press, Pittsburgh, 2000, 215.
59. Viggiano, J. A. S., Modeling the color of multi-colored halftones, in *TAGA Proceedings*, 1990, 44–62.
60. Clapper, F. R. and Yule, J. A. C., The effect of multiple internal reflections on the densities of half-tone prints on paper, *J. Optical Soc. Am.*, 43(7), 600–603, 1953.
61. Ruckdeschel, F. R. and Hauser, O. G., Yule–Nielsen effect in printing: a physical analysis, *Appl. Optics*, 17(21), 3376–3383, 1978.
62. Rogers, G. L., Optical dot gain in a halftone print, *J. Imaging Sci. Technol.*, 41(6), 643–656, 1997.
63. Rogers, G. L., A generalized Clapper–Yule model of halftone reflectance, *Color Res. Appl.*, 25(6), 403–407, 2000.
64. Gustavson, S., Color gamut of halftone reproduction, *J. Imaging Sci. Technol.*, 41(3), 283–290, 1997.

65. Gustavson, S., Dot Gain in Colour Halftones, Ph.D. thesis No. 492, Linköping University, Linköping, Sweden, September 1997.
66. Oittinen, P. and Saarelma, H., Influence of optical surface properties of paper on information capacity, *Paperi ja Puu — Paper and Timber*, 75(1–2), 66–71, 1993.
67. Arney, J. S., Probability description of the Yule–Nielsen effect, I, *J. Imaging Sci. Technol.*, 41(6), 633–636, 1997.
68. Arney, J. S., Wu, T., and Blehm, C., Modeling the Yule–Nielsen effect on color halftones, *J. Imaging Sci. Technol.*, 42(4), 335–340, 1998.
69. Hanrahan, P. and Krueger, W., Reflection from layered surfaces due to subsurface scattering, *Computer Graphics SIGGRAPH Proc.*, Anaheim, CA, August 1–6, 1993, 165–174.
70. Emmel, P. and Hersch, R. D., A unified model for color prediction of halftoned prints, *J. Imaging Sci. Technol.*, 44(4), 351–359, 2000.
71. Emmel, P., Modèles de Prédiction Couleur Appliqués à l'Impression Jet d'Encre, Ph.D. thesis No. 1857, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, 1998, 114, http://diwww.epfl.ch/w3lsp/publications/colour/thesis-emmel.html (in French).

chapter four

Color management for digital imaging systems

Edward J. Giorgianni
Thomas E. Madden
Kevin E. Spaulding
Eastman Kodak Company

Contents

4.1 Introduction

All successful color imaging systems employ some form of *color management*. Color management can be defined as a means for predicting, controlling, and adjusting color information throughout the system — from the initial color capture to the formation and display of output images. In chemical and other analog-based imaging systems, color management may be implemented in various ways, including equipment calibration, chemical process control, and operator-controlled or automated color-printing adjustments. In digital imaging systems, color management is generally implemented using software designed specifically for that purpose. The principal function of that software is to process (*transform*) image signals derived from an input device to make them appropriate for a given output device. Digital color management can be relatively simple when applied to imaging systems that are restricted to only certain types of inputs and outputs, but, when applied to systems having a variety of different types of input and output devices and media, color management can become quite complex.

The successful implementation of digital color management depends on a number of factors, including the use of appropriate *device characterization* methods and suitable mathematical techniques for forming and applying image-processing transformations. More fundamental to the success of the color management, however, are the selection of an appropriate *color management paradigm* and the use of a correspondingly appropriate method for *color encoding*, i.e., a method for representing color in digital form throughout the imaging process.

4.2 Color management paradigms

Underlying every color management approach is an implicitly or explicitly defined *paradigm* — an underlying conceptual model that ultimately determines how an imaging system using that color management will behave. The paradigm describes the expected relationships among the input images, encoded images, and output images of the system. Although various types of color imaging systems might behave quite differently, virtually all can be described in terms of just three fundamental types of color management paradigms. These paradigms will be referred to as Types *A*, *B*, and *C*.[1]

Color imaging systems based on a Type *A* color management paradigm are "input driven." Their color encoding represents the colors of the input images, and the colors produced by their outputs match (as much as possible) the input image colors. Color copiers, for example, operate according to a Type *A* paradigm; the normal expectation is that an output image produced by the copier will match the image input for copying. If an intermediary image (such as a video preview) is provided, it too would be expected to match the input and output images. This generally is the paradigm that first comes to people's minds when they think about color management. In fact, because the paradigm specifies that colors will match throughout an imaging system, the paradigm might seem to be the only one

that is needed. However, in many ways, the basic concept of the Type *A* paradigm is quite limited, which is why many commercial systems instead are based on Type *B* or Type *C* paradigms.

Systems based on a Type *B* color management paradigm are "encoding driven." Their color encoding is based on a unifying color encoding concept that tends to reduce or eliminate the colorimetric differences inherent in the system inputs. For example, some electronic prepress systems encode color in terms of the colorimetric characteristics of a reference reflection-print medium. Colors scanned from actual reflection prints are encoded essentially in terms of their measured colorimetry. But colors scanned from photographic transparency films are *re-rendered*, i.e., their measured colorimetric values are altered such that they correspond more closely to those that typically would be measured from the reference reflection-print medium. The current International Color Consortium (ICC) color management system is also based on a Type *B* paradigm in that all input images must be re-rendered to correspond to the properties of a reference imaging medium. As in a Type *A* paradigm system, the colors produced by the outputs of a Type *B* paradigm system are expected to visually match the colors represented by the color encoding. However, unlike a Type *A* paradigm system, the colors produced by a Type *B* paradigm system's outputs do not necessarily match the input image colors.

Systems based on a Type *C* color management paradigm are "output driven." Like Type *B* systems, they are based on a unifying color encoding concept. However, their output colors do not necessarily match the colors represented by this encoding, because additional re-rendering is performed, subsequent to encoding, as part of the output signal processing. This deliberate additional re-rendering might be done for simulation, i.e., to make one output produce images that imitate the appearance of images normally produced by another type of output. Re-rendering also might be done to enhance output images by taking advantage of the particular capabilities of each output device or medium. For example, when an output medium having a large color gamut is used, the output signal processing might include some expansion of the gamut of the encoded colors so as to use the full capabilities of that particular medium. This paradigm is often used in digital photofinishing systems where the objective is for each output to produce the best image possible from the encoded data. As a consequence of the output-specific re-renderings and color enhancements that might be performed, images produced on different types of output devices and media generally will not (by design) match each other.

These three paradigms are sufficient for describing the basic functionality of all existing types of color-managed imaging systems. Each paradigm is widely used, and each is technically valid. Yet each produces very different color results. The most appropriate paradigm for a given system will depend on the specific application for which that system will be used. It is also possible to design systems that function according to a *Universal Paradigm*, in which various input and output signal processing options are provided.[1]

Through the selection of appropriate options, such systems can be made to operate according to any of the described paradigms.

4.3 Digital color encoding

In addition to the selection of a color management paradigm appropriate for a given application, the successful implementation of digital color management requires the use of an appropriate method for digitally encoding color. The basic function of the digital color encoding is to provide a digital representation of colors for image processing, storage, and interchange among systems. Within a given color imaging system, the encoding provides a digital link between the system's inputs and outputs.

In a simple system, having just one type of input and one type of output, color encoding can be performed prior to any signal processing. The encoding is therefore a direct representation of the color values measured by the input device. In more complex systems supporting multiple types of inputs and outputs, such an arrangement is impractical, because each combination of input and output would require a separate signal-processing transform. For example, a single-output system requires two different transforms to process color values measured by two different input devices. The number of required system transforms in this arrangement equals the product of the number of inputs and outputs. Thirty-two signal-processing transforms are required, for example, in a system having four inputs and eight outputs.

A much more efficient system results if the color signal processing is split into two parts — input signal processing and output signal processing. In this arrangement, each input and each output has its own associated transform. Each input signal processing transform converts input color-signal values to values for a defined *color encoding specification*, and each output transform converts values from the color encoding specification to values appropriate for the particular output. In this arrangement, the number of system transforms equals just the sum, rather than the product, of the number of inputs and outputs. For example, only 12 signal-processing transforms are required in a system having 4 inputs and 8 outputs.

The success of this approach depends on the use of an appropriate color encoding specification. The specification must allow for color information to be represented unambiguously and in a way that does not limit the desired functionality of the system. A complete color encoding specification must define two principal attributes of the color representation: a *color encoding method* and a *color encoding data metric*. The color encoding method determines the actual meaning of the encoded data, while the color encoding data metric defines the color space and the numerical units in which encoded data are expressed. Some considerations involved in the design of a color encoding data metric will be discussed later.

The selection of the encoding method for a given system must be based on some color property — a particular aspect of color — that all of the inputs of that system have in common. It is that aspect of color that must be

measured and digitally encoded so as to represent color completely and unambiguously in the encoding specification. Three fundamental types of measurement and encoding methods are discussed in the following section.

4.4 Color encoding methods

Densitometric color encoding is based on input-image color measurements made according to defined sets of *spectral responsivities* that are not equivalent to any set of visual color-matching functions. The responsivities can be those of a particular type of densitometric instrument, such as an ISO *Status A* or *Status M densitometer*.[2,3] The responsivities also can be those of an actual scanner or of some hypothetical reference scanner. Encoded colors can be expressed in terms of red, green, and blue (RGB) densities, transmittances, or reflectances; cyan, magenta, and yellow (CMY) or cyan, magenta, yellow, and black (CMYK) colorant amounts; or other values associated with the densitometric measurements. The principal advantage of this type of encoding is that it corresponds quite directly to physical measurements of input images. Therefore, transformations from scanner RGB values to densitometric values, and transformations from densitometric values to output device RGB values, generally are quite simple. That simplicity can translate into optimum signal-processing accuracy and speed. However, the use of densitometric color encoding generally is limited to situations where all system input data are derived from essentially the same input medium. This often is the case in graphic arts and motion picture applications.

Colorimetric color encoding is similar to densitometric color encoding, except that it is derived from measurements made according to the spectral responsivities of a human observer. One of the principal advantages of this method is that it is based on well-established CIE recommendations for color measurement. At first glance, colorimetric encoding would seem to offer the perfect "device-independent" method for encoding color; in practice, colorimetric encoding sometimes can be used successfully where methods based on other forms of measurements will not work. Consider, for example, a system that supports input from an assortment of reflection media with image-forming colorants — printing inks, photographic dyes, thermal-transfer dyes, etc. — having different spectral absorption characteristics. A color encoding method based on RGB densitometric measurements alone would not provide a meaningful representation of color in this system. For example, a pair of colors on two different media might look identical, but they might produce quite different RGB densitometric values. Conversely, a pair of colors on two different media might appear quite different from one another, but they might happen to produce the same RGB densitometric values. These inconsistencies occur because the spectral absorption characteristics of the colorants used in the two media are different. Visual matches therefore will be *metameric*, not spectral. This makes colorimetric measurement a logical choice for color encoding. By definition, metameric pairs of color stimuli will have equal colorimetric values.

It is important to remember, however, that metameric matching is viewing-illuminant dependent. So, areas of color on different media that match when viewed under one illuminant might not match when viewed under another. This means that color encoding based on standard CIE colorimetric measurements can be used to encode color from multiple reflection media, but only if two conditions are realized. First, a single reference illuminant used for metameric matching must be specified, and, second, the encoded colorimetric values must be determined according to the spectral power distribution of that reference illuminant.

A further limitation of color encoding based on standard colorimetry alone is that it will not work for images input from media designed to be viewed under different conditions. For example, reflection prints generally are designed to be viewed under typical indoor conditions, while photographic slides are designed to be projected and viewed in a darkened room. Because an observer's perceptions will be affected by the differences in these respective viewing conditions, the colorimetric properties of reflection-print and projection-slide media must be fundamentally different. The colorimetric values measured from one type of medium will not be appropriate for use on the other. For example, if the colorimetry of a slide is measured and reproduced exactly on a reflection print, that print will appear too dark, too high in luminance contrast, and too cyan-blue in color balance. The use of colorimetric encoding therefore must be limited to media designed for one set of viewing conditions.

This is a serious problem, because three fundamentally different types of viewing environments are involved in the color imaging process, and there are limitless possible sets of viewing conditions for each of those types. On the input side of an imaging system, there are *original-scene environments*, i.e., the environments in which live original scenes are viewed and captured. Also, on the input side, there are *input-image environments*, where hardcopy and soft-copy images that are to be input to a color imaging system are viewed. Finally, there are *output-image environments,* where hardcopy and soft-copy images produced by a color imaging system eventually are viewed.

One means for dealing with the effects of various viewing conditions is *color appearance encoding*. In this technique, colorimetric values associated with one set of viewing conditions are transformed to determine a visually corresponding set of colorimetric values associated with another set of viewing conditions. The transformations, which are based on models of the human visual system, can account for differences in factors such as absolute image luminance level, image surround, and the observer's state of chromatic adaptation. Such transformations can be used, for example, to determine the colorimetric values required for a slide projected in a darkened room to visually match a reflection print viewed in a graphic arts viewing booth. Although transformations based on a color appearance model could be used quite directly in this particular example, other types of transformations are needed when an image is to be transformed from one *image state* to another, as described in the next section.

4.5 Image states

In a digital color imaging system, images can exist in several fundamentally different states. The image state is a function of how an image was captured, as well as any processing that may have been applied to the image. Although the concept of an image state can be applied to all attributes of an image, such as sharpness or noise, the aspect of image state that is of importance for the current discussion relates to the interpretation of the color values of the image. For example, the color values (digital code values) of an image could correspond to the sensor RGB values from a digital camera, the CIELAB values of a reflection print, or the ISO Status M RGB density values of a photographic negative. These image examples vary in two respects. First, they each use a different *color space* to encode the image (sensor RGB, CIELAB, and ISO Status M densities). However, just as importantly, each of these image encodings corresponds to a distinctly different *image state* (an original scene, a reflection print, and a photographic negative). Even if the same color space, CIELAB for example, were used to encode all of these images, it would still not be possible (or at least not optimal) to treat the images identically. Obviously, something quite different would have to be done with the CIELAB values of a color negative relative to the CIELAB values of a print. As will be discussed in more detail later, the same also would be true for the CIELAB values of an original scene relative to those of a print.

Most digital images can be broadly categorized into two types of image states: *unrendered* and *rendered*. Images in an unrendered image state are directly related to the colorimetry of real or hypothetical original scenes. Such images are sometimes called *scene-referred* images. Images in this category would include raw digital camera captures and images stored in the *Kodak PhotoYCC color interchange space*.[4] Images in a rendered image state are representations of the colorimetry of output images (such as a print, a slide, or a CRT display) and are sometimes called output-referred images. Many common color encodings, such as sRGB[5] and SWOP CMYK,[6] fall into this category. A third category of image state applies to the encoding of photographic color negatives. Unprocessed images captured by photographic color negative film scanners are in this image state, although it typically is a temporary state prior to forming a rendered image or determining a corresponding scene-referred image.

To enable the optimal use of digital images, it is important to distinguish images in an output-referred image state from those in a scene-referred image state. It is well known that the colorimetry of a pleasing rendered image generally does not match the colorimetry of the corresponding scene. Among other things, the tone/color reproduction process that "renders" the colors of a scene to the desired colors of the output image must compensate for differences between the scene and rendered image viewing conditions.[1,7] For example, rendered images generally are viewed at luminance levels much lower than those of typical outdoor scenes. Consequently, an increase in the overall contrast of the rendered image usually is required to compen-

sate for perceived losses in reproduced luminance and chrominance contrast. Additional contrast increases in the shadow regions of the rendered image also are needed to compensate for the viewing flare associated with rendered-image viewing conditions.

Psychological factors, such as color memory and color preference, also must be considered in image rendering. For example, observers generally remember colors as being of higher purity than they originally were, and they typically prefer the reproductions of skies and grasses to be more colorful than they were in the original scene. The tone/color reproduction aims of well-designed imaging systems will account for such factors.[1,7]

Finally, the tone/color reproduction process also must account for the fact that the dynamic range of an output device or medium usually is substantially less than that of an original scene. It is therefore typically necessary to discard and/or compress some of the highlight and shadow information of the scene to fit within the dynamic range of the rendered output image. This is shown in Figure 4.1, which illustrates a typical backlit scene. In this example, the approximate scene colorimetry was determined from a scan of a color negative. The image on the left shows a rendering of the scene appropriate for the foreground information, and the image on the right shows a rendering of the scene appropriate for the background infor-

(a) (b)

Figure 4.1 ***(See color insert on page 430)*** Rendering of large dynamic range scene optimized for (a) foreground, and (b) background.

mation. In the first case, much of the highlight information was clipped by the rendering process. Likewise, in the second case, much of the shadow information was lost. This is illustrated further in Figure 4.2, which shows a histogram of the scene luminance data for the image shown in Figure 4.1. A conventional reflection print of this scene can reproduce only about six stops (1.8 log luminance units) of scene information within the dynamic range of the output medium. The indicated ranges show the subsets of the scene luminance information corresponding to the two images in Figure 4.1. It can be seen that only a portion of the total scene information is reproduced in either of the rendered images.

Because the colorimetry of scenes and their corresponding rendered images are intentionally and necessarily different, it would be ambiguous to represent images in both image states using the same color encoding specification. For example, if one were to send the CIELAB values for a particular image, with no information about whether the color values were original-scene color values or rendered-image color values, the recipient would not know what to do with the image values so as to make a good output image. If the CIELAB values were rendered color values appropriate for the output viewing environment, it simply would be necessary to determine the device code values needed to produce the specified colorimetry. However, if the color values corresponded to original-scene color values, it would be necessary to modify the image colorimetry by applying an appropriate tone/color reproduction transformation before producing the output image. Directly reproducing the scene colorimetry on an output image gen-

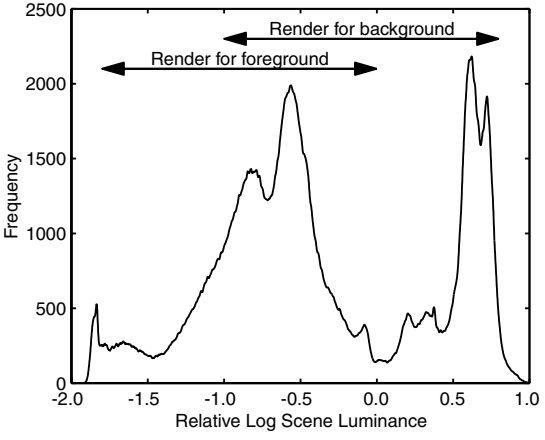


Figure 4.2 Histogram of relative log scene luminance values for the scene shown in Figure 4.1. A scene luminance range of about 1.8 log units can be reproduced on a typical output reflection print. Because the dynamic range of the original scene is substantially larger than this, a subset of the image data must be selected during the rendering process. Different results are obtained depending on whether the foreground or background region of the image is selected.

erally would produce results that would be judged inferior. For example, Figure 4.3 shows a pair of images generated from the same scene. Image (a) approximately matches the colorimetry of the original scene, whereas an appropriate tone/color reproduction transformation has been used to modify the colorimetry of image (b) to produce an image that generally would be judged to have improved color reproduction.

One of the advantages of encoding images in a scene-referred image state is that such encoding provides the capability of retaining the maximum amount of image information. As was illustrated in Figure 4.1, once an image is committed to a rendered output-referred image state appropriate for printing or display, any extended dynamic range information is permanently lost.

(a)

(b)

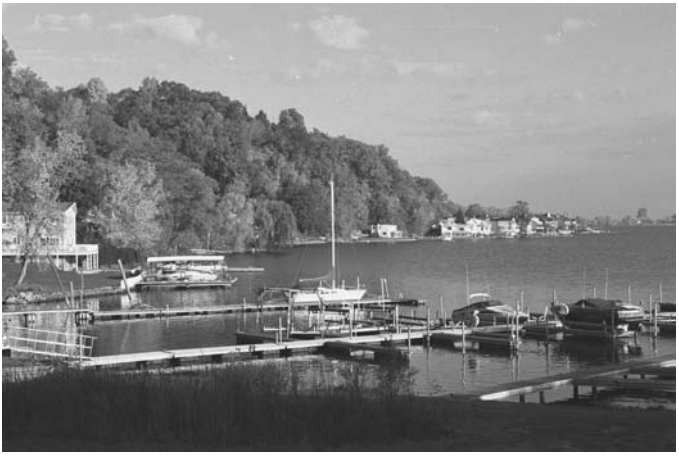


Figure 4.3 ***(See color insert)*** Two different renderings of a scene. In image (a), the colorimetry of the rendered image closely matches that of the original scene. In image (b), the rendered image is not colorimetrically accurate, but the resulting image generally would be judged to have improved color reproduction.

Retaining the scene-referred image data preserves the maximum flexibility for the potential uses of an image. This allows for the correction of image-capture exposure errors and enables multiple renditions to be made from a given image. For example, a photographer could decide at the time an image is printed whether to optimally render the foreground information or the background information from a backlit scene. It is valuable to preserve this option because there often will not be a single "best" choice that can be made when the image is captured. For the image shown in Figure 4.1, the final decision would depend on whether the photographer was most interested in the boys in the foreground or the scenic Alps in the background. Retaining the extended dynamic-range scene information also enables other options, such as employing advanced image processing techniques to produce a print wherein both the foreground and the background are well rendered, as shown in Figure 4.4. Comparable results could not be attained starting from one of the conventionally rendered images shown in Figure 4.1.

4.6 Standard image-state color encoding specifications

The fact that images exist in many different image states and are expressed in terms of numerous color spaces significantly complicates the development

Figure 4.4 ***(See color insert)*** In this rendering of the image shown in Figure 4.1, a digital "dodge-and-burn" operation has been used to produce a print in which both the foreground and the background are properly rendered.

of software applications that use and manipulate images. For example, an image-processing algorithm that works in one color space might not have the expected behavior when used in another color space. To reduce complexity of imaging system design, it is desirable to define standard color encodings for each of the main classes of image states. This provides for the unambiguous communication of color information and allows the development of standard image-manipulation algorithms and standard color-processing paths.

Attempts to standardize color encodings typically have involved the specification of a particular output-device-dependent color space that is central to the workflow for a certain market segment. Examples of such color spaces include sRGB and SWOP CMYK. Although such standardizations can work well within the limited scope of a particular application, significant compromises are necessary to use them in other applications. For example, hardcopy media and CRT displays typically have very different color gamuts. Therefore, using sRGB (which is based on a particular CRT model) as a standard color encoding necessarily involves clipping many colors that could have been produced on a given hardcopy medium. This would be unacceptable in many hardcopy-based market segments, such as consumer photo finishing and graphic arts.

The International Color Consortium (ICC) has defined a Profile Connection Space (PCS)[8] that comprises a color encoding specification that can be used to explicitly specify the color of an output-referred image with respect to a reference viewing environment. It could be argued that the PCS could serve as the standard color encoding specification for rendered images. However, it never was intended that the PCS be used to store or manipulate images directly. Rather, it was intended to be a color space where device profiles could be joined to form complete input-to-output color transforms. Neither the CIELAB nor the CIE XYZ color encodings supported for the PCS is particularly well suited for many common types of image manipulations. Additionally, quantization errors introduced by encoding images in PCS would be significantly larger than necessary, because a large percentage of code value combinations correspond to unrealizable colors.

Given the limitations of the existing solutions, Eastman Kodak Company has developed a family of color encoding specifications for use in the development of its digital imaging products.[9,10] These specifications are being offered for use by other companies, and they also have been proposed for international standardization. The following detailed discussion of the properties of these color encoding specifications will help to clarify several topics previously discussed, including color encoding methods, color encoding data metrics, and image states.

The first of these specifications, *Reference Input Medium Metric RGB* (*RIMM RGB*), is ideal for the manipulation, storage, and interchange of images from sources such as digital cameras that naturally capture *scene-referred* image data. A companion specification, *Reference Output Medium Metric RGB* (*ROMM RGB*), serves a similar purpose for images from sources such as print scanners and other devices that produce images in a rendered *output-referred*

image state. Figure 4.5 illustrates how these standard color encoding specifications can be used as the basis for a general imaging system architecture.

Before images can be sent to an output device, such as a printer, it generally will be necessary to convert scene-state images to rendered-state images using a tone/color rendering operation. However, in the same way that a negative is much more versatile than a print, an image in a scene-referred state will be much more versatile than one in a rendered-image state. Therefore, it is desirable in many imaging systems to delay any conversion to a rendered-image state until such time that an output image is to be generated. This provides the maximum flexibility for the imaging system.

4.6.1 *Criteria for selection of RIMM/ROMM RGB color encoding specifications*

It was desirable that the *RIMM RGB* and *ROMM RGB* color encoding specifications be defined such that they are as similar as possible to one another. Doing so simplifies the development of image-manipulation algorithms across the two color encodings. It also simplifies the rendering process in which a rendered *ROMM RGB* image is created from an original-scene image encoded in *RIMM RGB*. This desired similarity is best achieved by basing the data metrics of the two encoding specifications on the same color space. A number of criteria were used to select this color space. Specifically, the space should have the following properties:

- A direct relationship to the color appearance of the scene/image
- A color gamut large enough to encompass most real-world surface colors using non-negative tristimulus values
- An efficient encoding of the color information to minimize quantization artifacts
- A simple transformation to/from ICC PCS

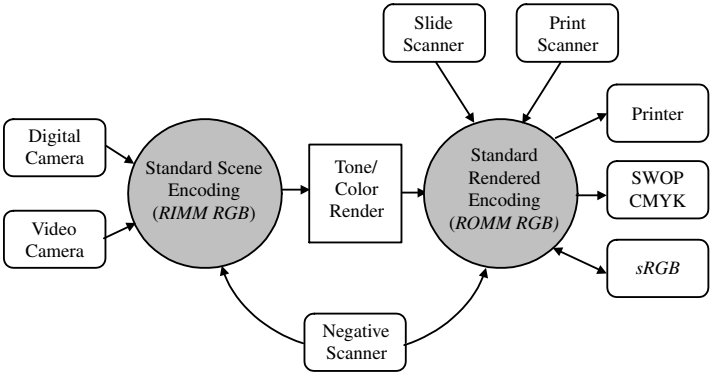


Figure 4.5 Image state diagram showing standard color encodings.

- A simple transformation to/from video RGB (e.g., sRGB)
- Be well suited for application of common image manipulations such as tone scale modifications, color-balance adjustments, sharpening, etc.
- Be compatible with established imaging workflows

All of these criteria can be achieved by the use of an additive RGB color space with an appropriately selected set of wide-gamut primaries. When images are encoded using a set of RGB primaries, there is a direct and simple relationship to scene/image colorimetry, because such primaries are linear transformations of the CIE XYZ primaries. RGB color spaces have the additional advantage that simple transformations based on a one-dimensional lookup table (LUT), a matrix, and another LUT can be used to convert to/from additive color spaces such as PCS XYZ, video RGB (sRGB), and digital camera RGB.

However, two of the criteria that affect the selection of the particular RGB primaries are somewhat conflicting. First, the chromaticities of the primaries should define a gamut sufficiently large to encompass colors likely to be found in real scenes and images. Such colors will then be defined by non-negative tristimulus values, which generally simplifies subsequent signal processing such as tone scale modifications. At the same time, their use should result in efficient digital encodings that minimize quantization errors.

Increasing the gamut to encompass more colors only can be achieved by trading off against correspondingly larger quantization errors (given a fixed bit depth). If the chromaticities of the primaries are chosen to include the maximum possible color gamut (for example, choosing the XYZ primaries would encompass the entire spectrum locus), a significant fraction of the color space would correspond to imaginary colors and to colors that would not commonly be encountered in real images. Therefore, in any encoding using such a color space, there would be large numbers of code value combinations that never would be used in practice. This would lead to larger quantization errors in the usable part of the color space than would be obtained with different primaries defining a smaller chromaticity gamut. It is, therefore, desirable to choose primaries with a gamut that is sufficiently large, but not larger than necessary.

Figure 4.6 shows the primaries selected for *RIMM/ROMM RGB*. These primaries encompass the gamut of real-world surface colors, without devoting a lot of space to non-realizable colors outside the spectrum locus. Also shown for comparison are the sRGB primaries. It can be seen that the area defined by the sRGB chromaticity boundaries is inadequate to cover significant portions of the real-world surface color gamut. In particular, it excludes many important high-chroma colors near the yellow-to-red boundary of the spectrum locus.

Another important requirement for the *RIMM RGB* and *ROMM RGB* color encoding specifications is that they be well suited for the application of common image manipulations. Many types of image manipulations

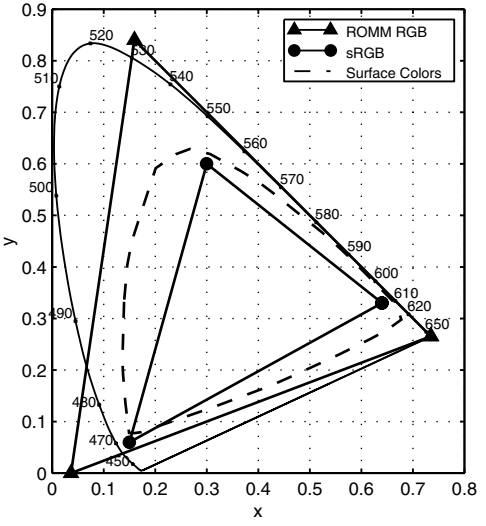


Figure 4.6 Comparison of *ROMM RGB* and sRGB primaries in *x-y* chromaticity coordinates.

include the step of applying nonlinear transformations to each of the channels of an RGB image (e.g., tone scale modifications, color balance adjustments, etc.). The process of forming a rendered image from a scene is one important application of this type. One way to accomplish the rendering operation is by the application of a nonlinear tone scale transformation to the individual channels of an RGB image in a scene-referred image state. A well-designed transformation of this type will have the desirable effects of increasing the luminance and color contrast in the mid-tones, compressing the contrast of the highlights and shadows, increasing the chroma of in-gamut colors, and gamut mapping out-of-gamut colors in a simple but visually pleasing way. If an input scene is represented using the *RIMM RGB* color encoding, the result of applying such rendering transforms will be a rendered image in the *ROMM RGB* color encoding.

Nonlinear channel-independent transforms will, in general, modify the ratios of the red, green, and blue channel data. This can lead to unwanted hue shifts, particularly for high-chroma colors. Hue shifts are particularly problematic in reproductions of natural chroma gradients having constant hue and saturation. Such gradients occur when rounded surfaces are illuminated by a moderately directional light source. In these situations, chroma increases with increasing distance from the specular highlight and then decreases again as the shadows deepen.

The induction of hue shifts by the application of the nonlinear channel-independent transforms can never be completely eliminated. One objective for optimizing the location of the primaries was to eliminate or minimize objectionable hue shifts, sometimes at the expense of less noticeable or less

likely hue shifts. Hue shifts for a particular color can be eliminated when the color lies on one of the straight lines passing through the primaries and the white point on a chromaticity diagram.

The effects of nonlinear transforms on hue shifts were studied using a chroma series for eight color patches from the Macbeth Color Checker. These patches included red, yellow, green, cyan, blue, magenta, light skin, and dark skin. Hue shifts in skin tones and yellows, particularly in the direction of green, are considered the most objectionable. These hue shifts are most strongly affected by the location of the blue primary. Other colors that were considered particularly important during the optimization process were blues and reds.

There is a trade-off between the color gamut of the primaries, quantization artifacts, and the extent of the hue shifts that occur during rendering. If the primaries are moved out to increase the color gamut, quantization artifacts will increase, and the hue shifts introduced during the application of a nonlinear transformation generally will decrease. This results from the fact that the RGB values in real images will be distributed over a smaller range, thereby reducing the impact of nonlinear transformations. If the color gamut is decreased by moving the primaries closer together, quantization artifacts diminish, but hue shifts are generally larger, and color gamut is sacrificed.

Finally, a basic requirement for any commercially useful color encoding is that it be compatible with typical commercial imaging workflows. In many cases, Adobe Photoshop software is an important component in such imaging chains. Conveniently, Adobe Photoshop versions 5.0 and higher have incorporated the concept of a "working color space," which is different from the monitor preview color space. This is consistent with the concept of storing/manipulating images in an extended color gamut space. Adobe has placed a constraint on the definition of valid working color spaces that requires the primaries to have all-positive x, y, and z chromaticity values. This condition is satisfied for the *ROMM RGB* primaries.† (Because Adobe Photoshop software operates within a rendered-image paradigm, it is inappropriate to use *RIMM RGB* as a Photoshop software working color space.)

During the selection of the *RIMM/ROMM RGB* primaries, an extensive optimization process was used to determine the best overall solution to satisfy all of these criteria. The CIELAB hue shifts associated with the selected *RIMM/ROMM RGB* primaries are shown in Figure 4.7. This plot shows a series of line segments connecting corresponding pairs of CIELAB a^*, b^* values, before and after a nonlinear tone scale transformation was applied to a chroma series in each of eight color directions. It can be seen that only relatively small hue shifts are introduced for the highest chroma colors in the blue and cyan directions, and the hue shifts elsewhere are virtually negligible. Overall, these hue shifts are very small compared to those

† For more information about using *ROMM RGB* as a Photoshop software working space, see the white paper posted at www.kodak.com (search on "ROMM").

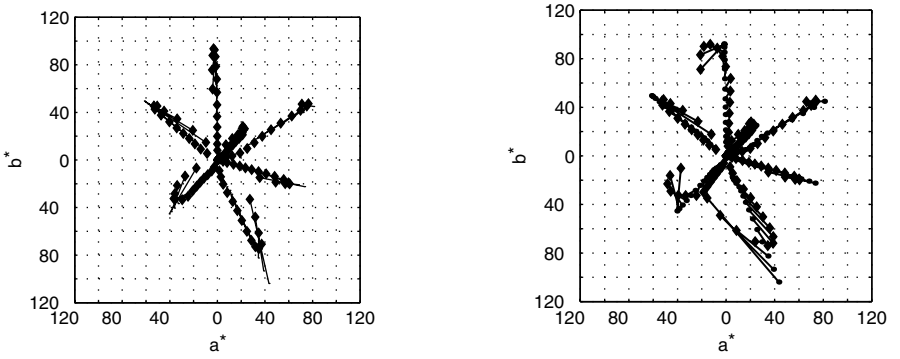


Figure 4.7 Hue shifts resulting from a typical nonlinear rendering transform for (a) the *RIMM/ROMM RGB* primaries, and (b) an alternate set of wide-gamut primaries. The hue shifts for the most important colors are visually negligible for the *RIMM/ROMM RGB* color encoding.

associated with most other sets of additive primaries. Similar results were obtained when this hue-shift analysis was carried out using several other color spaces, including CIECAM97s,[11] IPT,[12] and the OSA_UCS color space.[13]

4.6.2 ROMM RGB color encoding specification

Reference Output Medium Metric RGB (*ROMM RGB*) is designed to be an extended-gamut color encoding specification for representing the color appearance of an output-referred image. In addition to specifying the image state and color space, it is also necessary to specify an intended viewing environment to define unambiguously an encoding of color appearance. One of the requirements for *ROMM RGB* is that it be tightly coupled to the ICC Profile Connection Space (PCS). Color values in the PCS represent the CIE colorimetry of a defined reference medium that will produce the desired color appearance when viewed in a reference viewing environment. The reference viewing environment for *ROMM RGB* was based on that defined in the latest ICC draft specification[14] and is specified to have the following characteristics:

1. The luminance level for the observer adaptive white is 160 cd/m^2.
2. The observer adaptive white has the chromaticity values of CIE Standard Illuminant D_{50} ($x = 0.3457$, $y = 0.3585$).
3. The viewing surround is average, i.e., the overall luminance level and chromaticity of the surround are assumed to be similar to be that of the image.
4. There is 0.75% viewing flare, referenced to the observer adaptive white.

5. The image color values are assumed to be encoded using flareless (or flare-corrected) colorimetric measurements based on the CIE 1931 Standard Colorimetric Observer.

The *ROMM RGB* color encoding is defined in the context of a reference imaging medium associated with a hypothetical additive color device having the following characteristics:

1. Reference primaries defined by the CIE chromaticities given in Table 4.1
2. Equal amounts of the reference primaries producing a neutral with the chromaticity of D_{50} ($x = 0.3457$, $y = 0.3585$)
3. The capability of producing a white with a luminance factor of $F_W = 0.89$ and a black with a luminance factor of $F_K = 0.0030911$

Table 4.1 Primaries/White Point for Reference Imaging Medium

Color	x	y
Red	0.7347	0.2653
Green	0.1596	0.8404
Blue	0.0366	0.0001
White	0.3457	0.3585

Images intended to be viewed in other viewing environments, or on a medium different from the reference medium, can be encoded in *ROMM RGB* by first determining the corresponding tristimulus values that would produce the intended color appearance on the reference medium when viewed in the reference viewing environment. The corresponding tristimulus values can be determined by using appropriate color appearance transformations to account for the differences between the actual and reference viewing conditions. Additionally, it may be necessary to account for differences in the media characteristics.

The conversion of the PCS XYZ tristimulus values to *ROMM RGB* values can be performed by a matrix operation followed by a set of one-dimensional functions. This is equivalent to the operations associated with a basic monitor profile, which means that *ROMM RGB* can be incorporated in a system employing ICC profiles simply by using an appropriately designed display profile.

Most current implementations of the ICC PCS incorporate the concept of a reference medium wherein the black point of the reference medium is mapped to $Y_{PCS} = 0$, and the white point of the reference medium is mapped

to $Y_{PCS} = 1.0$.[15] Therefore, to relate actual CIE image colorimetry to PCS XYZ values, an appropriate normalizing transformation is required as follows:

$$X_{PCS} = \frac{(X - X_K)}{(X_W - X_K)} \frac{X_W}{Y_W}$$

$$Y_{PCS} = \frac{(Y - Y_K)}{(Y_W - Y_K)}$$

$$Z_{PCS} = \frac{(Z - Z_K)}{(Z_W - Z_K)} \frac{Z_W}{Y_W} \tag{4.1}$$

where X, Y, Z = CIE image tristimulus values
$X_{PCS}, Y_{PCS}, Z_{PCS}$ = PCS tristimulus values
X_W, Y_W, Z_W = tristimulus values of the reference medium white point ($X_W = F_W X_0 = 85.81$, $Y_W = F_W Y_0 = 89.00$, and $Z_W = F_W Z_0 = 73.42$, where $X_0 = 96.42$, $Y_0 = 100.00$, and $Z_0 = 82.49$)
X_K, Y_K, Z_K = tristimulus values of the reference medium black point ($X_K = F_K X_0 = 0.2980$, $Y_K = F_K Y_0 = 0.3091$, and $Z_K = F_K Z_0 = 0.2550$)

4.6.2.1 ROMM RGB conversion matrix

Given the defined primaries shown in Table 4.1, the following matrix can be derived to compute the linear *ROMM RGB* values from the PCS image tristimulus values:

$$\begin{bmatrix} R_{ROMM} \\ G_{ROMM} \\ B_{ROMM} \end{bmatrix} = \begin{bmatrix} 1.3460 & -0.2556 & -0.0511 \\ -0.5446 & 1.5082 & 0.0205 \\ 0.0000 & 0.0000 & 1.2123 \end{bmatrix} \begin{bmatrix} X_{PCS} \\ Y_{PCS} \\ Z_{PCS} \end{bmatrix} \tag{4.2}$$

As required by the definition of *ROMM RGB*, this matrix will map image tristimulus values with the chromaticity of D_{50} to equal *ROMM RGB* values. A neutral with a Y_{PCS} value of 1.0, corresponding to the reference medium white point, will map to linear *ROMM RGB* values of 1.0. Likewise, the reference medium black point will map to linear *ROMM RGB* values of 0.0.

4.6.2.2 Nonlinear encoding of ROMM RGB

A nonlinear quantization function is used to store the *ROMM RGB* values in integer form. A simple gamma function nonlinearity incorporating a slope limit at the dark end of the intensity scale is defined for this purpose.

$$C'_{ROMM} = \begin{cases} 0; & C_{ROMM} < 0.0 \\ 16C_{ROMM}I_{max}; & 0.0 \le C_{ROMM} < E_t \\ C_{ROMM}^{1.1.8}I_{max}; & E_t \le C_{ROMM} < 1.0 \\ I_{max}; & C_{ROMM} \ge 1.0 \end{cases} \quad (4.3)$$

where C is R, G, or B; $I_{\max}$ = the maximum integer value used for the nonlinear encoding; and

$$E_t = 16^{1.8(1-1.8)} = 0.001953 \quad (4.4)$$

For the baseline 8-bit configuration, $I_{\max}$ is equal to 255. The linear segment of the nonlinearity is used to impose a slope limit to eliminate reversibility problems that otherwise would result from the infinite slope of the gamma function at the zero point. Twelve-bit and 16-bit versions of *ROMM RGB* are also defined. The only difference is that the value of $I_{\max}$ is set to 4095 or 65535, respectively. In cases where it is necessary to identify a specific precision level, the notations *ROMM8 RGB*, *ROMM12 RGB*, and *ROMM16 RGB* are used. Table 4.2 shows some sample encodings for a series of neutral patches of specified Y_{PCS}.

Table 4.2 Sample Neutral Patch Encodings

Y_{PCS}	*ROMM8 RGB*	*ROMM12 RGB*	*ROMM16 RGB*
0.00	0	0	0
0.001	4	66	1049
0.01	20	317	5074
0.10	71	1139	18236
0.18	98	1579	25278
0.35	142	2285	36574
0.50	174	2786	44590
0.75	217	2490	55855
1.00	255	4095	65535

4.6.3 *RIMM RGB color encoding specification*

Reference Input Medium Metric RGB (*RIMM RGB*) is a companion color encoding specification to *ROMM RGB* that can be used to encode the colorimetry of an *unrendered scene*. Both encodings utilize the same wide-gamut color space defined by the primaries and white point given in Table 4.1. The reference viewing conditions used to encode scene color

values for *RIMM RGB* are typical of outdoor environments and are defined as follows:

- The luminance level for the observer adaptive white is 15,000 cd/m^2.
- The observer adaptive white has the chromaticity values of CIE Standard Illuminant D_{50} ($x = 0.3457$, $y = 0.3585$).
- Viewing surround is average, i.e., the overall luminance level and chrominance of the surround is assumed similar to that of the scene.
- There is no viewing flare for the scene other than that already included in the scene colorimetric values.
- The scene color values are assumed to be encoded using flareless (or flare corrected) colorimetric measurements based on the CIE 1931 Standard Colorimetric Observer.

Scenes captured under conditions different from those of the reference viewing environment can be encoded in *RIMM RGB* by first determining the corresponding tristimulus values that would produce the intended color appearance in the reference viewing environment. For some applications, the intended color appearance may be an estimate of the appearance of the original scene if it had been captured in the reference viewing environment. For other applications, it may be desirable to encode the color appearance of the scene in its particular capture viewing environment. In this case, corresponding tristimulus values can be determined by using appropriate color appearance transformations to account for the differences between the actual and reference viewing conditions.

4.6.3.1 RIMM RGB conversion matrix

Because *ROMM RGB* and *RIMM RGB* use a common color space, the conversion from scene tristimulus values to corresponding linear *RIMM RGB* values can be accomplished using the same conversion matrix that was given in Equation 4.2, except that the input tristimulus values are scene XYZ values rather than PCS XYZ values.

$$\begin{bmatrix} R_{RIMM} \\ G_{RIMM} \\ B_{RIMM} \end{bmatrix} = \begin{bmatrix} 1.3460 & -0.2556 & -0.0511 \\ -0.5446 & 1.5082 & 0.0205 \\ 0.0000 & 0.0000 & 1.2123 \end{bmatrix} \begin{bmatrix} X_{D50} \\ Y_{D50} \\ Z_{D50} \end{bmatrix} \tag{4.5}$$

Note: The scene *XYZ* values are normalized such that the luminance of a correctly exposed perfect white diffuser in the scene will have a value of $Y_{D50} = 1.0$.

4.6.3.2 Nonlinear encoding of RIMM RGB

Because the dynamic range of unrendered scenes is generally larger than that of the medium specified for *ROMM RGB*, a different nonlinear encoding must be used. The *RIMM RGB* nonlinearity is based on that specified by

Recommendation ITU-R BT.709[16] (formerly known as CCIR 709). This is the same nonlinearity used in the *Kodak PhotoYCC color interchange space* encoding implemented in the *Kodak Photo CD system*[4] and is given by

$$C'_{RIMM} = \begin{cases} 0; & C_{RIMM} < 0.0 \\ \frac{I_{max}}{V_{clip}} 4.5 C_{RIMM}; & 0.0 \le C_{RIMM} < 0.018 \\ \frac{I_{max}}{V_{clip}} 1.099 C_{RIMM}^{0.45} - 0.099; & 0.018 \le C_{RIMM} < E_{clip} \\ I_{max}; & C_{RIMM} \ge E_{clip} \end{cases} \tag{4.6}$$

where C is either R, G, or B; I_{max} is the maximum integer value used for the nonlinear encoding; $E_{clip} = 2.0$ is the normalized scene luminance level that is mapped to I_{max}; and

$$V_{clip} = 1.099 E_{clip}^{0.45} - 0.099 = 1.402 \tag{4.7}$$

For the baseline 8-bit/channel *RIMM RGB* configuration, I_{max} is 255. In some applications, it may be desirable to use a higher-bit-precision version of *RIMM RGB* to minimize any quantization errors. Twelve-bit and 16-bit per channel versions of *RIMM RGB* are also defined. The only difference is that the value of I_{max} is set to 4095 or 65535, respectively. In cases in which it is necessary to identify a specific precision level, the notations *RIMM8 RGB*, *RIMM12 RGB,* and *RIMM16 RGB* are used.

4.6.4 ERIMM RGB color encoding specification

The *RIMM RGB* color space is defined to have an extended luminance dynamic range that can encode information up to 200% of the luminance value associated with a normally exposed perfect (100%) diffuse white reflector in the scene. This should be adequate for many input sources, such as digital cameras, which themselves have a somewhat limited dynamic range. However, for some inputs, most notably scanned photographic negatives, a greater luminance dynamic range is required to encode the full range of captured scene information. For example, consider the histogram of scene luminance data previously shown in Figure 4.2. The *RIMM RGB* encoding would only retain scene information up to a log relative scene luminance value of 0.3. A significant portion of the scene information would be lost with a *RIMM RGB* encoding in this case. To provide an encoding that can retain the full range of captured scene information, a variation of the *RIMM RGB* color space, *Extended Reference Input Medium Metric RGB (ERIMM RGB)*, is defined.

As with *RIMM RGB*, *ERIMM RGB* is related directly to the colorimetry of an original scene. The nonlinear encoding function is the only encoding operation that is different. For *ERIMM RGB*, it is desirable to increase both the maximum scene luminance value that can be represented as well as to reduce the quantization interval size. The size of the quantization interval is directly related to the minimum scene luminance value that can be accurately represented. To satisfy both the extended luminance dynamic range and reduced quantization interval requirements simultaneously, it is necessary to use a greater bit precision for *ERIMM RGB*. A minimum of 12 bits per color channel is recommended.

4.6.4.1 *Nonlinear encoding for ERIMM RGB*

A modified logarithmic encoding is used for *ERIMM RGB*. A linear segment is included for the very lowest luminance values to eliminate the noninvertibility of a strictly logarithmic encoding at the dark end of the scale. The encoding was defined such that the linear and logarithmic segments match in both value and derivative at the boundary. In equation form, this encoding is represented by

$$C'_{ERIMM} = \begin{cases} 0; & C_{RIMM} \le 0 \\ \left(\dfrac{0.0789626}{E_t}\right) C_{RIMM}\, I_{\max}; & 0 < C_{RIMM} \le E_t \\ \left(\dfrac{\log C_{RIMM} + 3.0}{5.5}\right) I_{\max}; & E_t < C_{RIMM} \le E_{clip} \\ I_{\max}; & C_{RIMM} > E_{clip} \end{cases} \tag{4.8}$$

where C is R, G, or B; I_{max} is the maximum integer value used for the nonlinear encoding; $E_{clip} = 10^{2.5} = 316.23$ = the upper scene luminance limit that gets mapped to I_{max}; and

$$E_t = e/1000 = 0.00271828 \tag{4.9}$$

is the break point between the linear and logarithmic segments, e being the base of the natural logarithm. For a 12-bit encoding, I_{max} is 4095, and for a 16-bit encoding, I_{max} is 65535. In cases in which it is necessary to identify a specific precision level, the respective notations *ERIMM12 RGB* and *ERIMM16 RGB* are used.

To compute *ERIMM RGB* values, Equation 4.8 should be used in place of Equation 4.6 in the previously described procedure for determining *RIMM RGB* values. Examples of *RIMM RGB* and *ERIMM RGB* encodings for neutral patches at different scene relative luminance levels are shown in Table 4.3. It can be seen that the range of relative luminances that can be represented in *ERIMM RGB* is greatly extended from that of *RIMM RGB*.

Table 4.3 Sample Scene Luminance Encodings

Relative Luminance	Relative Log Luminance	RIMM8 RGB	RIMM12 RGB	ERIMM12 RGB
0.001	–3.00	1	13	119
0.01	–2.00	8	131	745
0.10	–1.00	53	849	1489
0.18	–0.75	74	1194	1679
1.00	0.00	182	2920	2234
2.00	0.30	255	4095	2458
8.00	0.90	NA	NA	2906
32.00	1.50	NA	NA	3354
316.23	2.50	NA	NA	4095

4.7 *Image states in a color managed architecture*

The use of color management systems, such as that developed by the ICC, is becoming increasingly common in a variety of digital imaging applications. Color management systems typically are based on an architecture in which the color response of an input device is characterized using an *input profile,* which describes the relationship between the device code values and color values in some profile connection space (PCS). Similarly, the color response of an output device is characterized using an *output profile,* which describes the relationship between the PCS color values and the corresponding device code values needed to produce colors having those values. The PCS used in the ICC color management architecture, and the color encodings used in virtually every other color management system, are defined to be in a rendered output-referred image state. This type of PCS, generally based on reflection-print media viewed in indoor viewing environments, greatly complicates the use of a color management architecture based on the image-states paradigm. For example, a traditional input profile cannot be used for an input device that captures scene-referred data if it is desired to convert the image data to the standard scene-referred color encoding (i.e., *(E)RIMM RGB*). This is because the output of such a profile would be PCS color values in a rendered-image state. As discussed previously, the process of rendering an image from a scene-image state to a rendered-image state typically will involve an irreversible loss of information. Thus, transforming an image into *RIMM RGB* by combining a device-to-PCS profile with a PCS-to-*RIMM RGB* profile would seriously compromise the quality of the resulting image.

However, this does not mean that traditional color management architectures must be discarded altogether to build an imaging system around the image-state paradigm previously shown in Figure 4.5. Rather, it simply means that conventional input/output profiles cannot be used in the imaging

chain until the point at which the image is ready to be committed to a final output rendering. Fortunately, most color management systems provide for the concept of a *device link profile* that can be used to bypass the PCS and go directly from an input color space to an output color space. (Typically, such device link profiles would be created by cascading an input profile with an output profile, but this is not a requirement.)

Figure 4.8 illustrates this approach in more detail. Device link profiles are used to transform scene-referred input images into *RIMM RGB*. Input-image sources might include digital cameras (when unrendered sensor RGB values are available), as well as color negative film scanners (when special transformations are used to extract scene colorimetry from scanned densitometric values). In this case, not only is *RIMM RGB* used as a stopping point where images can be stored or edited, but it also becomes the output color space for the device profiles, effectively serving the role of a "scene-referred profile connection space." At the point when it is finally desired to produce an image on an output device such as a printer or CRT, a conventional input profile can be used to render the *RIMM RGB* image to the PCS. This profile would include the desired system tone/color reproduction characteristics. A conventional output profile then can be used to transform the PCS image to the appropriate output device code values.

Conventional input profiles can be used for input devices, such as print/slide scanners and CRTs, where the input images already are in a rendered image state. These input profiles can be combined directly with output profiles to produce an image for a particular output device. Alternatively, the input profile can be combined with a *ROMM RGB* profile to convert the image to *ROMM RGB* for the purposes of storage, interchange, or editing. Because *ROMM RGB* is a simple LUT/matrix away from ICC

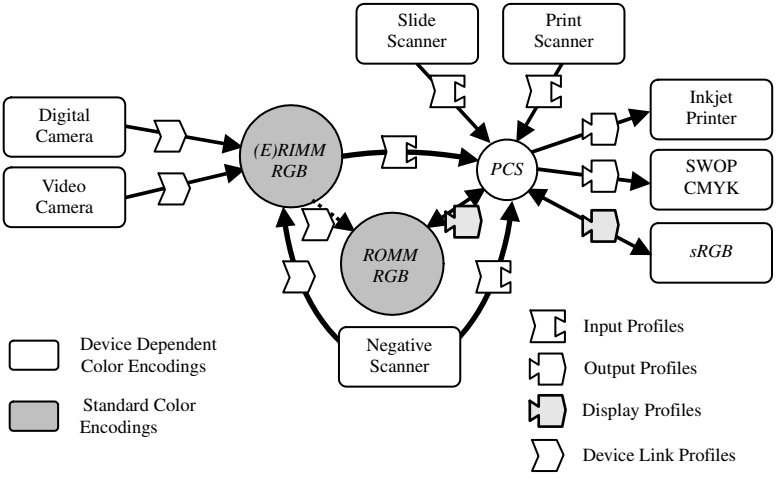


Figure 4.8 Image state architecture using color management.

PCS XYZ values, it falls within the class of color encodings that can be represented with a simple display profile.

It should be noted that the input profiles used for rendered images intended to be viewed in environments significantly different from the reference viewing environment defined for the PCS must also include appropriate viewing environment transformations. For example, photographic slides are typically intended to be viewed in a darkened room. The colorimetric characteristics of slide films are designed with a higher luminance contrast so as to produce pleasing images in that viewing environment. Therefore, an input profile for a slide scanner must not only account for the colorimetric characteristics of the scanner; it must also include an appropriate transformation that will determine the visually equivalent colorimetry for the PCS reference viewing environment. It may also be necessary for the profile to perform some amount of *re-rendering* of the image to map the extended dynamic range of the slide film into the reflection-print-like dynamic range of the PCS reference medium.

During the process of working with images that are stored in the *RIMM/ROMM RGB* color encodings, it frequently will be desirable to preview the image on a video display. In a color-managed system, this can be accomplished by combining the appropriate *RIMM RGB* or *ROMM RGB* profile with a display profile for the particular video display. Because *RIMM/ROMM RGB* are based on a simple additive color space, a simple display-type profile using only a LUT followed by a matrix generally can be used to get to PCS XYZ. Likewise, the output profile for the video display would comprise a matrix followed by a gamma-function nonlinearity. For cases where processing speed is a critical concern, these operations can be combined, yielding a simple LUT–matrix–LUT processing chain that can be implemented directly and optimized for speed.

An example of an imaging chain for a representative system utilizing the standard image state architecture is shown in Figure 4.9. The input device for this example is a color negative film scanner. A device link profile is used to convert the raw film scanner image to a corresponding *ERIMM RGB* image. This profile accounts for the characteristics of the scanner as well as the characteristics of the film used to capture the image. Once the image is in *ERIMM RGB*, many different types of algorithms can be used to operate on the image. For example, a *scene balance algorithm* can be used to automatically color balance the image to correct for any variations in capture illumination and/or film processing, or an advanced tone scale algorithm could be used to properly darken the background of a backlit scene. *ERIMM RGB* is an appropriate color encoding for applying many types of image-processing algorithms, but it is especially important that algorithms utilizing the extended dynamic range scene information of the encoding be applied in *ERIMM RGB* before the image is rendered to an output-referred state.

After all scene-state image manipulations have been applied, the image can be rendered to produce a corresponding rendered-state image. In this example, the image is converted to a *ROMM RGB* representation where

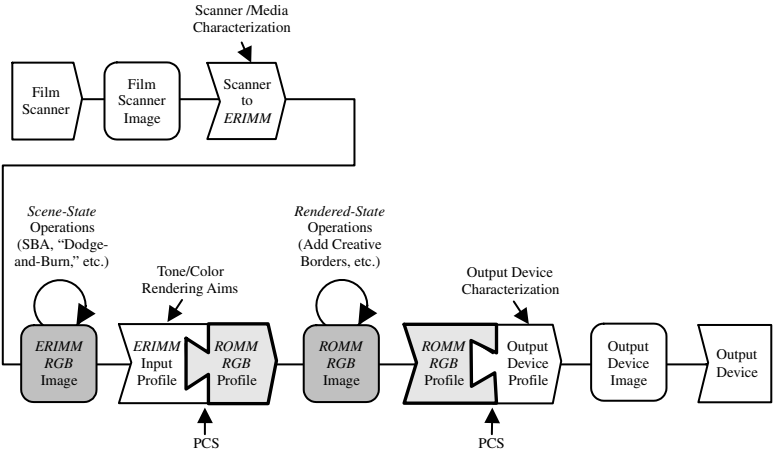


Figure 4.9 Imaging chain example using standard color encodings and color management.

further operations will be applied. This conversion can be applied by combining an *ERIMM RGB* input profile with a *ROMM RGB* profile. The *ERIMM RGB* input profile is used to impart the system tone/color reproduction aims relating the scene color values to the corresponding rendered image color values. These aims may be application dependent. For example, consumer photographers generally prefer higher contrast and higher saturation images than those preferred by professional portrait photographers. In many cases, acceptable tone/color reproduction characteristics can be achieved by applying a simple tone reproduction curve to the *ERIMM RGB* scene-exposure values. In this case, the *ERIMM RGB* to *ROMM RGB* transformation will involve only a simple one-dimensional LUT.

Once the image is in *ROMM RGB*, additional rendered-state image operations can be applied. For example, text annotations and a creative border could be added to the image, or the image could be composited with an image from a print scanner, etc. The final *ROMM RGB* image can then be printed by applying a *ROMM RGB* profile and an output profile for the particular output device.

4.8 Digital color management with JPEG 2000

Historically, many desktop imaging applications have been designed based on the assumption that the digital image stored in a file is ready to display directly on a CRT. This assumption has caused significant interoperability problems for applications that have attempted to store images with other color encodings. For example, if an application were to open a *ROMM RGB* image and send the color values directly to a video display, the image would appear very desaturated, because the image was encoded using a set of high-

chroma primaries rather than video primaries. Special software must be used to open and/or color manage images stored in various color spaces, and, as a result, images stored in spaces other than video RGB cannot be used by a large number of applications. The overall situation has effectively made it impractical to use color spaces other than video RGB for most consumer applications.

JPEG 2000 is a new file storage format that has been recently standardized. One of the requirements that have been built into the format specification is that all JPEG 2000 compliant file readers must be able to properly decode an image stored in any color encoding specification that can be defined using a restricted class of ICC profiles. In particular, the supported ICC profile formats include any display-type profile that utilizes a LUT-matrix transformation to get to PCS XYZ. Both the *(E)RIMM RGB* and the *ROMM RGB* color encoding specifications can be represented using profiles that fall within this definition. As a result, images can be stored using these color encoding specifications without sacrificing interoperability. Applications designed to manipulate images in the color spaces of these encodings will be able to do so. Other applications can simply use the attached ICC profile to convert the image to a video RGB color space (e.g., *sRGB*) or to some other color space for which the application was designed.

4.9 Summary

In digital imaging systems, the principal role of color management is to transform image signals derived from one or more input devices to signals that are appropriate for a given output device. Digital color management can be relatively straightforward when applied to simpler systems, but it becomes quite complex when applied to systems having a variety of different input and output types.

The successful implementation of digital color management depends on a number of factors, including the use of appropriate device characterization methods and suitable mathematical techniques for forming and applying image-processing transformations. In addition, an appropriate color management paradigm must be determined for the particular system being developed. Virtually all current color-managed imaging systems are based on one of three basic paradigms. A "universal" paradigm, in which various input and output signal processing options are supported, has also been defined. Through the selection of appropriate options, systems based on this all-inclusive paradigm can be made to operate according to any of the three basic paradigms.

Successful color management also requires the use of an appropriate method for encoding color. Three basic color encoding methods were described in this chapter. Densitometric color encoding is based on input-image color measurements made according to any of various defined sets of spectral responsivities. Colorimetric color encoding is derived from measurements made according to the spectral responsivities of a standard human

observer. Color appearance encoding is an extension of basic colorimetric encoding. In this method, colorimetric values associated with one set of viewing conditions are transformed to determine a visually corresponding set of colorimetric values associated with another set of viewing conditions. The transformations account for differences in a number of factors — including absolute image luminance level, image surround, and the observer's state of chromatic adaptation — that influence an observer's perception of color.

In addition to an appropriate color encoding method, a properly designed data metric must also be used. A data metric defines the color space and numerical units in which encoded data are expressed. The combination of a color encoding method and data metric forms a complete color encoding specification. The selected color encoding specification must be consistent with the state of the image to be encoded. The image state is a function of how an image was captured and subsequently signal processed. Most digital images can be categorized into two types of image states: unrendered and rendered. Images in an unrendered state are directly related to the colorimetry of real or hypothetical original scenes. Images in a rendered state are encoded representations of the colorimetry of output images. Transformations beyond those based on color appearance alone are needed when an image is to be transformed from one image state to another. Image-state transformations are greatly facilitated by the use of appropriate color encoding specifications.

Eastman Kodak Company has developed a family of such specifications for use in the development of its digital imaging products. These specifications have been proposed for international standardization. *Reference Input Medium Metric RGB (RIMM RGB)* is designed for the manipulation, storage, and interchange of images from sources that naturally capture scene-referred (unrendered) image data. *Reference Output Medium Metric RGB (ROMM RGB)* serves a similar purpose for images from sources that produce images in an output-referred (rendered) image state. Images encoded in terms of *RIMM RGB* or *ROMM RGB* are fully compliant with the JPEG 2000 file storage format.

References

1. Giorgianni, E. J. and Madden, T. E., *Digital Color Management: Encoding Solutions*, Addison-Wesley, Reading, MA, 1998.
2. ISO, Recommendation R5: Diffuse Transmission Density, Photography, 1955.
3. ISO, International Standard 5/3: Photography — Density Measurements — Part 3: Spectral Conditions, 1984.
4. *KODAK Photo CD System — A Planning Guide for Developers,* Eastman Kodak Company, Rochester, NY, 1991.
5. Multimedia Systems and Equipment — Colour Measurement and Management — Part 2–1: Colour Management — Default RGB Colour Space — sRGB, IEC 61966–2–1, 1999.

6. Graphic Technology — Color Characterization Data for Type 1 Printing, ANSI/CGATS TR 001, 1995.
7. Hunt, R. W. G., *The Reproduction of Colour*, 5th ed., Fountain Press, England, 1995.
8. International Color Consortium, *File Format for Color Profiles,* Specification ICC.1A: 1999–04.
9. Spaulding, K. E., Woolfe, G. J., and Giorgianni, E. J., Reference input/output medium metric RGB color encodings (*RIMM/ROMM RGB*), in *Proc. IS&T's 2000 PICS Conference,* 2000, 155–163.
10. Spaulding, K. E., Woolfe, G. J., and Giorgianni, E. J., Image states and standard color encodings *(RIMM/ROMM RGB),* in *Proc. Eighth Color Imaging Conference: Color Science and Engineering: Systems, Technologies, Applications,* 2000, 288–294.
11. Luo, M. R. and Hunt, R. W. G., The structure of CIE 1997 color appearance model (CIECAM97s), *Color Res. Appl*. 23, 138–146, 1998.
12. Ebner, F. and Fairchild, M., Development and testing of a color space (IPT) with improved hue uniformity, in *Proc. Sixth Color Imaging Conference,* 1998, 8–13.
13. Billmeyer, F. W., Survey of color order systems, *Color Res. Appl*. 12, 173–186, 1987.
14. L. Borg, The profile Connection Space, ICC votable proposal submission 19.10, June 8, 2000.
15. Interpretation of the PCS, appendix to Kodak ICC profile for CMYK (SWOP) input, ANSI CGATS/SC6 N 254, June 3, 1998.
16. Basic parameter values for the HDTV Standard for the Studio and for International Programme Exchange, Recommendation ITU-R BT.709 (formerly CCIR Recommendation 709).

chapter five

Device characterization

Raja Balasubramanian
Xerox Solutions & Services Technology Center

Contents

5.1 Introduction

Achieving consistent and high-quality color reproduction in a color imaging system necessitates a comprehensive understanding of the color characteristics of the various devices in the system. This understanding is achieved through a process of device characterization. One approach for doing this is known as closed-loop characterization, where a specific input device is optimized for rendering images to a specific output device. A common example of closed-loop systems is found in offset press printing, where a drum scanner is often tuned to output CMYK signals for optimum reproduction on a particular offset press. The tuning is often carried out manually by skilled press operators. Another example of a closed-loop system is traditional photography, where the characteristics of the photographic dyes, film, development, and printing processes are co-optimized (again, often manually) for proper reproduction. While the closed-loop paradigm works well in the aforementioned examples, it is not an efficient means of managing color in open digital color imaging systems where color can be exchanged among a large and variable number of color devices. For example, a system comprising three scanners and four printers would require $3 \times 4 = 12$ closed-loop transformations. Clearly, as more devices are added to the system, it becomes difficult to derive and maintain characterizations for all the various combinations of devices.

An alternative approach that is increasingly embraced by the digital color imaging community is the device-independent paradigm, where translations among different device color representations are accomplished via an intermediary device-independent color representation. This approach is more efficient and easily managed than the closed-loop model. Taking the same example of three scanners and four printers now requires only $3 + 4 = 7$ transformations. The device-independent color space is usually based on a colorimetric standard such as CIE XYZ or CIELAB. Hence, the visual system is explicitly introduced into the color imaging path. The closed-loop and device-independent approaches are compared in Figure 5.1.

The characterization techniques discussed in this chapter subscribe to the device-independent paradigm and, as such, involve deriving transformations between device-dependent and colorimetric representations. Indeed, a plethora of device characterization techniques have been reported in the literature. The optimal approach depends on several factors, including the physical color characteristics of the device, the desired quality of the characterization, and the cost and effort that one is willing to bear to perform the characterization. There are, however, some fundamental concepts that are common to all these approaches. We begin this chapter with a description of these concepts and then provide a more detailed exposition of characterization techniques for commonly encountered input and output devices. To keep the chapter to a manageable size, an exhaustive treatment is given to only a few topics. The chapter is complemented by an extensive set of references for a more in-depth study of the remaining topics.

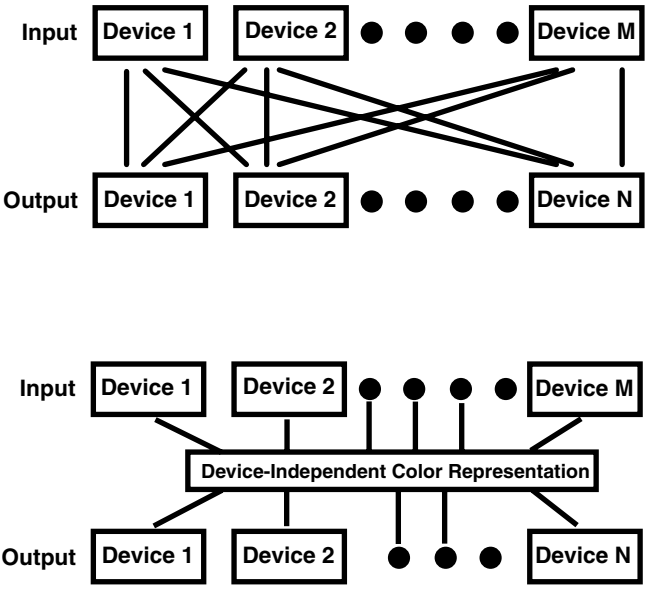


Figure 5.1 Closed-loop vs. device-independent color management.

5.2 Basic concepts

It is useful to partition the transformation between device-dependent and device-independent space into a calibration and a characterization function, as shown in Figure 5.2.

5.2.1 Device calibration

Device calibration is the process of maintaining the device with a fixed known characteristic color response and is a precursor to characterization. Calibration can involve simply ensuring that the controls internal to the device are kept at fixed nominal settings (as is often the case with scanners and digital cameras). Often, if a specific color characteristic is desired, this typically requires making color measurements and deriving correction functions to ensure that the device maintains that desired characteristic. Sometimes the desired characteristic is defined individually for each of the device signals; e.g., for a CRT display, each of the R, G, B channels is often linearized with respect to luminance. This linearization can be implemented with a set of one-dimensional tone reproduction curves (TRCs) for each of the R, G, B signals. Sometimes, the desired characteristic is defined in terms of mixtures of device signals. The most common form of this is gray-balanced calibration, whereby equal amounts of device color signals (e.g., R = G = B or C = M = Y) correspond to device-independent measurements that are neutral or gray

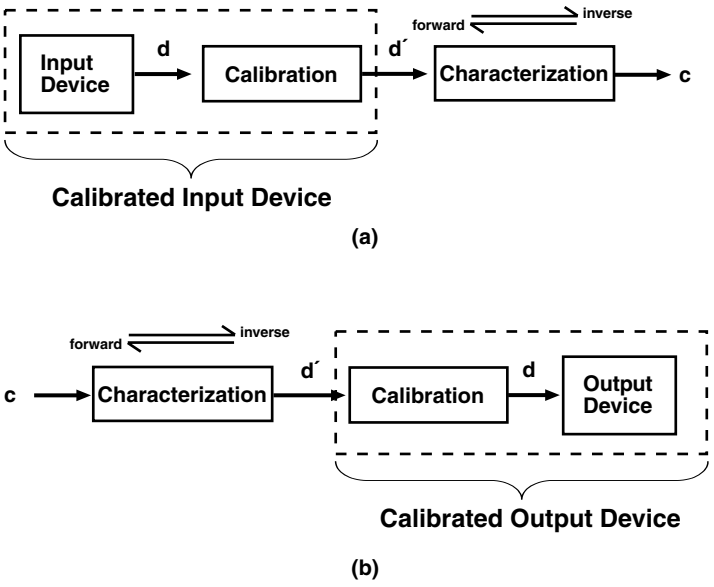


Figure 5.2 Calibration and characterization for input and output devices.

(e.g., a*= b*= 0 in CIELAB coordinates). Gray-balancing of a device can also be accomplished with a set of TRCs.

It is important to bear mind that calibration with one-dimensional TRCs can control the characteristic response of the device only in a limited region of color space. For example, TRCs that ensure a certain tone response along each of the R, G, B axes do not necessarily ensure control of the gray axis, and vice versa. However, it is hoped that this limited control is sufficient to maintain, within a reasonable tolerance, a characteristic response within the entire color gamut; indeed, this is true in many cases.

5.2.2 Device characterization

The characterization process derives the relationship between device-dependent and device-independent color representations for a calibrated device. For input devices, the captured device signal is first processed through a calibration function (see Figure 5.2) while output devices are addressed through a final calibration function. In typical color management workflows, device characterization is a painstaking process that is done infrequently, while the simpler calibration process is carried out relatively frequently to compensate for temporal changes in the device's response and maintain it in a fixed known state. It is thus assumed that a calibrated device maintains the validity of the characterization function at all times. Note that calibration and characterization form a pair, so that if a new calibration alters the characteristic color response of the device, the characterization must also be re-derived.

The characterization function can be defined in two directions. The forward characterization transform defines the response of the device to a known input, thus describing the color characteristics of the device. The inverse characterization transform compensates for these characteristics and determines the input to the device that is required to obtain a desired response. The inverse function is used in the final imaging path to perform color correction to images.

The sense of the forward function is different for input and output devices. For input devices, the forward function is a mapping from a device-independent color stimulus to the resulting device signals recorded when the device is exposed to that stimulus. For output devices, this is a mapping from device-dependent colors driving the device to the resulting rendered color, in device-independent coordinates. In either case, the sense of the inverse function is the opposite to that of the forward function.

There are two approaches to deriving the forward characterization function. One approach uses a model that describes the physical process by which the device captures or renders color. The parameters of the model are usually derived with a relatively small number of color samples. The second approach is empirical, using a relatively large set of color samples in conjunction with some type of mathematical fitting or interpolation technique to derive the characterization function. Derivation of the inverse function calls for an empirical or mathematical technique for inverting the forward function. (Note that the inversion does not require additional color samples; it is purely a computational step.)

A primary advantage to model-based approaches is that they require fewer measurements and are thus less laborious and time consuming than empirical methods. To some extent, a physical model can be generalized for different image capture or rendering conditions, whereas an empirical technique is typically optimized for a restrictive set of conditions and must be rederived as the conditions change. Model-based approaches generate relatively smooth characterization functions, whereas empirical techniques are subject to additional noise from measurements and often require additional smoothing on the data. However, the quality of a model-based characterization is determined by the extent to which the model reflects the real behavior of the device. Certain types of devices are not readily amenable to tractable physical models; thus, one must resort to empirical approaches in these cases. Also, most model-based approaches require access to the raw device, while empirical techniques can often be applied in addition to simple calibration and characterization functions already built into the device. Finally, hybrid techniques can be employed that borrow strengths from both model-based and empirical approaches. Examples of these will be presented later in the chapter.

The output of the calibration and characterization process is a set of mappings between device-independent and -dependent color descriptions; these are usually implemented as some combination of power-law mapping, 3×3 matrix conversion, white-point normalization, and one-dimensional and multidimensional lookup tables. This information can be stored in a

variety of formats, of which the most widely adopted industry standard is the International Color Consortium (ICC) profile (www.color.org). For printers, the Adobe Postscript language (Level 2 and higher) also contains operators for storing characterization information.[1]

It is important to bear in mind that device calibration and characterization, as described in this chapter, are functions that depend on color signals alone and are not functions of time or the spatial location of the captured or rendered image. The overall accuracy of a characterization is thus limited by the ability of the device to exhibit spatial uniformity and temporal stability. Indeed, in reality, the color characteristics of any device will vary to some degree over its spatial footprint and over time. It is generally good practice to gather an understanding of these variances prior to or during the characterization process. This may be accomplished by exercising the device response with multiple sets of stimuli in different spatial orientations and over a period of time. The variation in the device's response to the same stimulus across time and space is then observed. A simple way to reduce the effects of nonuniformity and instability during the characterization process is to average the data at different points in space and time that correspond to the same input stimulus.

Another caution to keep in mind is that many devices have color-correction algorithms already built into them. This is particularly true of low-cost devices targeted for consumers. These algorithms are based in part on calibration and characterization done by the device manufacturer. In some devices, particularly digital cameras, the algorithms use spatial context and image-dependent information to perform the correction. As indicated in the preceding paragraph, calibration or characterization by the user is best performed if these built-in algorithms can be deactivated or are known to the extent that they can be inverted. (This is especially true of the model-based approaches.) Reverse engineering of built-in correction functions is not always an easy task. One can also argue that, in many instances, the built-in algorithms provide satisfactory quality for the intended market, hence not requiring additional correction. Device calibration and characterization is therefore recommended only when it is necessary and possible to fully control the color characteristics of the device.

5.2.3 *Input device calibration and characterization*

There are two main types of digital color input devices: scanners, which capture light reflected from or transmitted through a medium, and digital cameras, which directly capture light from a scene. The captured light passes through a set of color filters (most commonly, red, green, blue) and is then sensed by an array of charge-coupled devices (CCDs). The basic model that describes the response of an image capture device with M filters is given by

$$D_i = \int_{\lambda \varepsilon V} S(\lambda) q_i(\lambda) u(\lambda) \partial\lambda + n_i, i = 1, \ldots, M \tag{5.1}$$

where D_i = sensor response
$S(\lambda)$ = input spectral radiance
$q_i(\lambda)$ = spectral sensitivity of the ith sensor
$u(\lambda)$ = detector sensitivity
n_i = measurement noise in the ith channel
V = spectral regime outside which the device sensitivity is negligible

Digital still cameras often include an infrared (IR) filter; this would be incorporated into the $u(\lambda)$ term. Invariably, $M = 3$ sensors are employed with filters sensitive to the red, green, and blue portions of the spectrum. The spectral sensitivities of a typical set of scanner filters are shown in Figure 5.3. Scanners also contain an internal light source that illuminates the reflective or transmissive material being scanned. Figure 5.4 shows the spectral radiance of a fluorescent scanner illuminant. Note the sharp spikes that typify fluorescent sources. The light incident upon the detector is given by

$$S(\lambda) = I_s(\lambda)R(\lambda) \tag{5.2}$$

where $R(\lambda)$ = spectral reflectance (or transmittance) function of the input stimulus
$I_s(\lambda)$ = scanner illuminant

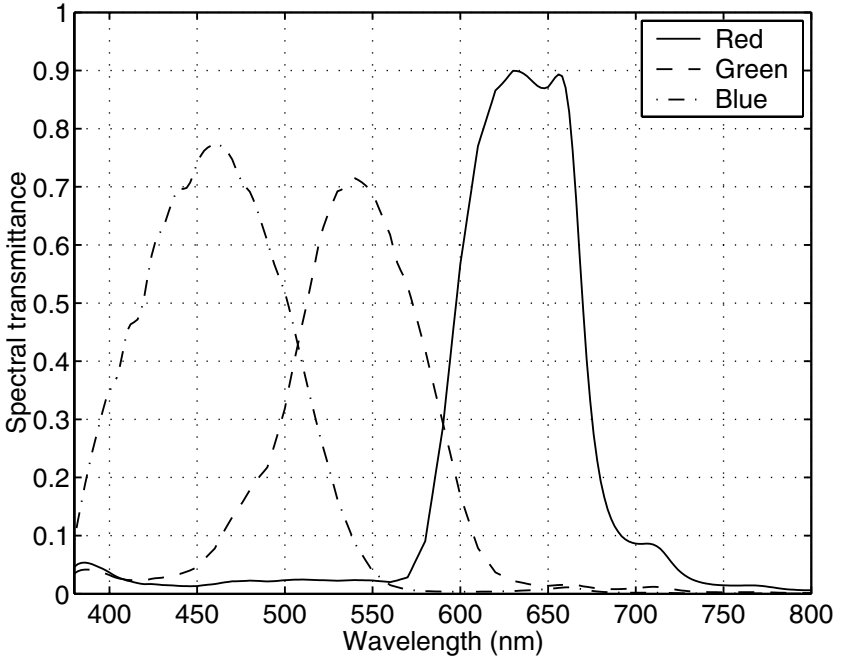


Figure 5.3 Typical scanner filter sensitivities.

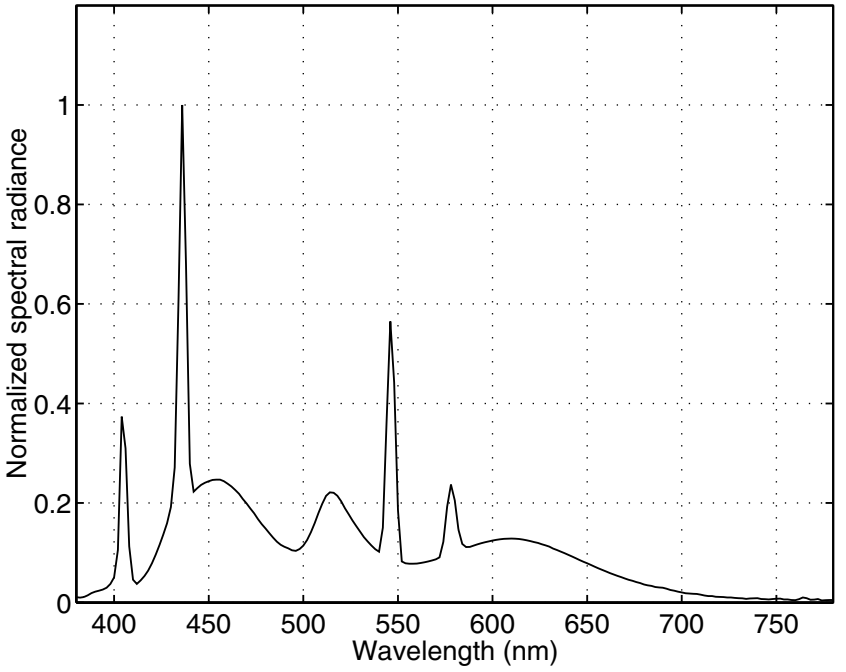


Figure 5.4 Spectral radiance of typical scanner illuminant.

From the introductory chapter on colorimetry, we know that spectral radiance is related to colorimetric signals by

$$C_i = K_i \int_{\lambda \varepsilon V} S(\lambda) c_i(\lambda) \partial\lambda, i = 1,2,3 \tag{5.3}$$

where C_i = colorimetric signals
$c_i(\lambda)$ = corresponding color matching functions
K_i = normalizing constants

Again, if a reflective sample is viewed under an illuminant $I_v(\lambda)$, the input spectral radiance is given by

$$S(\lambda) = I_v(\lambda) R(\lambda) \tag{5.4}$$

Equations 5.1 through 5.4 together establish a relationship between device-dependent and device-independent signals for an input device. To further explore this relationship, let us represent a spectral signal by a discrete L-vector comprising samples at wavelengths $\lambda_1, \ldots, \lambda_L$. Equation 5.1 can be rewritten as

$$\mathbf{d} = A_d^t \mathbf{s} + \varepsilon \tag{5.5}$$

where $\mathbf{d}$ = M-vector of device signals
$\mathbf{s}$ = L-vector describing the input spectral signal
$\mathbf{A}_d$ = $L \times M$ matrix whose columns are the input device sensor responses
ε = noise term

If the input stimulus is reflective or transmissive, then the illuminant term $I_s(\lambda)$ can be combined with either the input signal vector $\mathbf{s}$ or the sensitivity matrix $\mathbf{A}_d$. In a similar fashion, Equation 5.3 can be rewritten as

$$\mathbf{c} = \mathbf{A}_c^t \mathbf{s} \tag{5.6}$$

where $\mathbf{c}$ = colorimetric three-vector
$\mathbf{A}_c$ = $L \times 3$ matrix whose columns contain the color-matching functions $c_i(\lambda)$

If the stimulus being viewed is a reflection print, then the viewing illuminant $I_v(\lambda)$ can be incorporated into either $\mathbf{s}$ or $\mathbf{A}_c$.

It is easily seen from Equations 5.5 and 5.6 that, in the absence of noise, a unique mapping exists between device-dependent signals $\mathbf{d}$ and device-independent signals $\mathbf{c}$ if there exists a transformation from the device sensor response matrix $\mathbf{A}_d$ to the matrix of color matching functions $\mathbf{A}_c$.[2] In the case of three device channels, this translates to the condition that $\mathbf{A}_d$ must be a linear nonsingular transformation of $\mathbf{A}_c$.[3,4] Devices that fulfill this so-called *Luther–Ives* condition are referred to as *colorimetric* devices.

Unfortunately, practical considerations make it difficult to design sensors that meet this condition. For one thing, the assumption of a noise-free system is unrealistic. It has been shown that, in the presence of noise, the Luther–Ives condition is not optimal in general, and it guarantees colorimetric capture only under a single viewing illuminant I_v.[5] Furthermore, to maximize the efficiency, or signal-to-noise ratio (SNR), most filter sets are designed to have narrowband characteristics, as opposed to the relatively broadband color matching functions. For scanners, the peaks of the R, G, B filter responses are usually designed to coincide with the peaks of the spectral absorption functions of the C, M, Y colorants that constitute the stimuli being scanned. Such scanners are sometimes referred to as *densitometric* scanners. Because photography is probably the most common source for scanned material, scanner manufacturers often design their filters to suit the spectral characteristics of photographic dyes. Similar observations hold for digital still cameras, where filters are designed to be narrowband, equally spaced, and independent so as to maximize efficiency and enable acceptable shutter speeds. A potential outcome of this is scanner metamerism, where two

stimuli that appear identical to the visual system may result in distinct scanner responses, and vice versa.

The spectral characteristics of the sensors have profound implications on input device characterization. The narrowband sensor characteristics result in a relationship between XYZ and device RGB that is typically more complex than a 3×3 matrix, and furthermore changes as a function of properties of the input stimulus (i.e., medium, colorants, illuminant). A colorimetric filter set, on the other hand, results in a simple linear characterization function that is media independent and that does not suffer from metamerism. For these reasons, there has been considerable interest in designing filters that approach colorimetric characteristics, subject to practical constraints that motivate the densitometric characteristics.[6] An alternative approach is to employ more than three filters to better approximate the spectral content of the input stimulus.[7] These efforts are largely in the research phase; most input devices in the market today still employ three narrowband filters. Hence, the most accurate characterization is a nonlinear function that varies with the input medium.

Model-based characterization techniques use the basic form of Equation 5.1 to predict device signals D_i given the radiance $S(\lambda)$ of an arbitrary input medium and illuminant, and the device spectral sensitivities. The latter can sometimes be directly acquired from the manufacturer. However, due to temporal changes in device characteristics and variations from device to device, a more reliable method is to estimate the sensitivities from measurements of suitable targets. Model-based approaches may be used in situations where there is no way of determining *a priori* the characteristics of the specific stimulus being scanned. However, the accuracy of the characterization is directly related to the accuracy of the model and its estimated parameters. The result is usually an $M \times 3$ matrix that maps M (typically three) device signals to three colorimetric signals such as XYZ.

Empirical techniques, on the other hand, directly correlate colorimetric measurements of a color target with corresponding device values that result when the device is exposed to the target. Empirical techniques are suitable when the physical nature of the input stimulus is known beforehand, and a color target with the same physical traits is available for characterizing the input device. An example is the use of a photographic target to characterize a scanner that is expected to scan photographic prints. The characterization can be a complex nonlinear function chosen to achieve the desired level of accuracy, and it is obtained through an empirical data-fitting or interpolation procedure.

Modeling techniques are often used by researchers and device manufacturers to better understand and optimize device characteristics. In end user applications, empirical approaches are often adopted, as these provide a more accurate characterization than model-based approaches for a specific set of image capture conditions. This is particularly true for the case of scanners, where it is possible to classify *a priori* a few commonly encountered media (e.g., photography, lithography, xerography, inkjet) and generate

empirical characterizations for each class. In the case of digital cameras, it is not always easy to define or classify the type of stimuli to be encountered in a real scene. In this case, it may be necessary to revert to model-based approaches that assume generic scene characteristics. More details will be presented in following sections.

A generic workflow for input device characterization is shown in Figure 5.5. First, the device is calibrated, usually by ensuring that various internal settings are in a fixed nominal state. For scanners, calibration minimally involves normalizing the RGB responses to the measurement of a built-in white tile, a process that is usually transparent to the user. In addition, it may be desirable to linearize and gray-balance the device response by scanning a suitable premeasured target. Next, the characterization is performed using a target comprising a set of color patches that spans the gamut of the input medium. Often, the same target is used for both linearization and characterization. Industry standard targets designed for scanners are the Q60 and IT8. Device-independent color measurements are made of each patch in the target using a spectroradiometer, spectrophotometer, or colorimeter. Additional data processing may be necessary to extract raw colorimetric data from the measurements generated by the instrument. Next, the input device records an image of the target. If characterization is being performed as a separate step after calibration, then the captured image must be processed through the calibration functions derived in a previous step. The device-dependent (typically RGB) coordinates for each patch on the target must then be extracted from the image. This involves correctly identifying the spatial extent of each patch within the scanned image. To facilitate this, it is desirable to include reference fiducial marks at each corner of the target and supply target layout information (e.g., number of rows, columns) to the image-processing software. Also, it is recommended that a subset of pixels near the center of each patch is averaged, so as to reduce the effect of spatial noise in the device response. Once extracted, the device-dependent values are correlated with the corresponding device-independent values to obtain the characterization for the device.

The forward characterization is a model of how the device responds to a known device-independent input; i.e., it is a function that maps device-

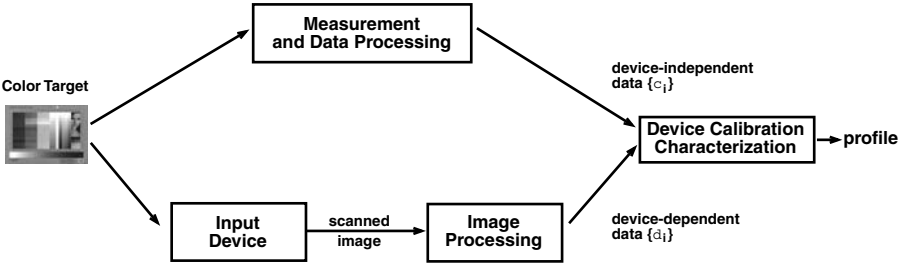


Figure 5.5 Input device characterization workflow.

independent measurements to the resulting device signals. The inverse function compensates for the device characteristics and maps device signals to corresponding device-independent values. Model-based techniques estimate the forward function, which is then inverted using analytic or numerical approaches. Empirical techniques derive both the forward and inverse functions.

Figure 5.6 describes how the accuracy of the resulting characterization can be evaluated. A test target containing colors that are preferably different from those in the initial characterization target is presented to the image-capture device. The target should be made with the same colorants and media as used for the characterization target. The resulting captured electronic image is mapped through the same image-processing functions performed when the characterization was derived (see Figure 5.5). It is then converted to a device-independent color space using the inverse characterization function. The device-independent color values of the patches are then extracted and compared with measurements of these patches using an appropriate color difference formula such as ΔE^*_{ab} or ΔE^*_{94} (described in more detail in Section 5.5). To avoid redundant processing, the same target can be used for both deriving and testing the characterization, with different portions of the target being used for the two purposes.

5.2.4 Output device calibration and characterization

Output color devices can be broadly categorized into emissive display devices and devices that produce reflective prints or transparencies. Emissive devices produce colors via additive mixing of red, green, and blue (RGB) lights. Examples are cathode ray tube (CRT) displays, liquid crystal displays (LCDs), organic light emitting diodes (OLEDs), plasma displays, projection displays, etc. The spectral radiance emitted by a display device is a function of the input digital RGB values and is denoted $S_{RGB}(\lambda)$. Two important assumptions are usually made that greatly simplify display characterization.

- *Channel independence.* Each of the R, G, B channels to the display operates independently of the others. This assumption allows us to separate the contribution of spectral radiance from the three channels.

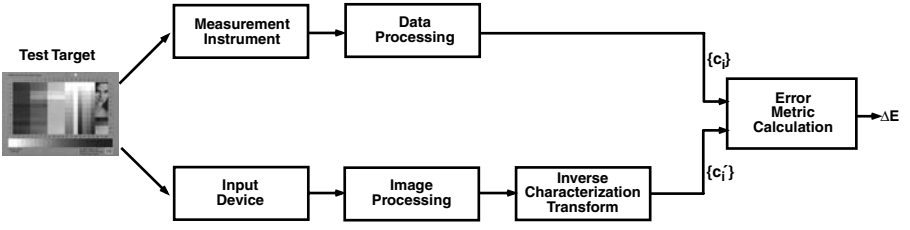


Figure 5.6 Testing of input device characterization.

$$S_{RGB}(\lambda) = S_R(\lambda) + S_G(\lambda) + S_B(\lambda) \quad (5.7)$$

- *Chromaticity constancy.* The spectral radiance due to a given channel has the same basic shape and is only scaled as a function of the device signal driving the display. This assumption further simplifies Equation 5.7 to

$$S_{RGB}(\lambda) = f_R(D_R)\, S_{Rmax}(\lambda) + f_G(D_G)\, S_{Gmax}(\lambda) + f_B(D_B)\, S_{Bmax}(\lambda) \quad (5.8)$$

where $S_{Rmax}(\lambda)$ = the spectral radiance emitted when the red channel is at its maximum intensity
D_R = the digital input to the display
$f_R()$ = a linearization function (discussed further in Section 5.8)

The terms for green and blue are similarly defined. Note that a constant scaling of a spectral radiance function does not change its chromaticity (*x*-*y*) coordinates, hence the term "chromaticity constancy."

These assumptions hold fairly well for many display technologies and result in a simple linear characterization function. Figure 5.7 shows the spectral radiance functions for a typical CRT phosphor set. Sections 5.8 and 5.9 contain more details on CRT and LCD characterization, respectively. Recent research has shown that OLEDs can also be accurately characterized with techniques similar to those described in these sections.[8]

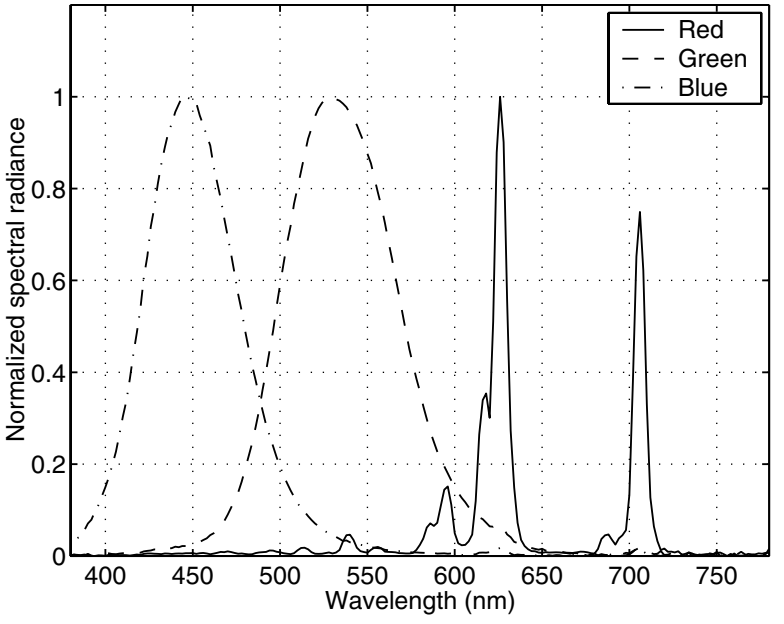


Figure 5.7 Spectral radiance of typical CRT phosphors.

Printing devices produce color via subtractive color mixing in which a base medium for the colorants (usually paper or transparency) reflects or transmits most of the light at all visible wavelengths, and different spectral distributions are produced by combining cyan, magenta, and yellow (CMY) colorants to selectively remove energy from the red, green, and blue portions of the electromagnetic spectrum of a light source. Often, a black colorant (K) is used both to increase the capability to produce dark colors and to reduce the use of expensive color inks. Photographic prints and transparencies and offset, laser, and inkjet printing use subtractive color.

Printers can be broadly classified as being continuous-tone or halftone devices. A continuous-tone process generates uniform colorant layers and modulates the concentration of each colorant to produce different intensity levels. A halftone process generates dots at a small fixed number of concentration levels and modulates the size, shape, and frequency of the dots to produces different intensity levels. (Color halftoning is covered in detail in another chapter.) Both types of processes exhibit complex nonlinear color characteristics, making them more challenging to model and characterize. For one thing, the spectral absorption characteristics of printed colorants do not fulfill the ideal "block dye" assumption, which states that the C, M, Y colorants absorb light in nonoverlapping bands in the long, medium, and short wavelengths, respectively. Such an ideal behavior would result in a simple linear characterization function. Instead, in reality, each of these colorants exhibits unwanted absorptions in other bands, as shown in Figure 5.8, giving rise to complex intercolorant interactions and nonlinear characterization functions. Halftoning introduces additional optical and spatial interactions and thus lends complexity to the characterization function. Nevertheless, much effort has been devoted toward the modeling of continuous and halftone printers as well as toward empirical techniques. A few of these techniques will be explored in further detail in Section 5.10.

A generic workflow for output device calibration and characterization is given in Figure 5.9. A digital target of color patches with known device values is sent to the device. The resulting displayed or printed colors are measured in device-independent (or colorimetric) color coordinates, and a relationship is established between device-dependent and device-

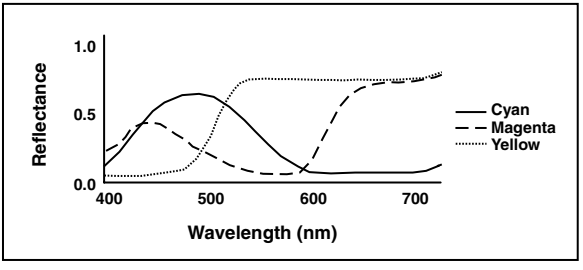


Figure 5.8 Spectral absorption functions of typical C, M, Y colorants.

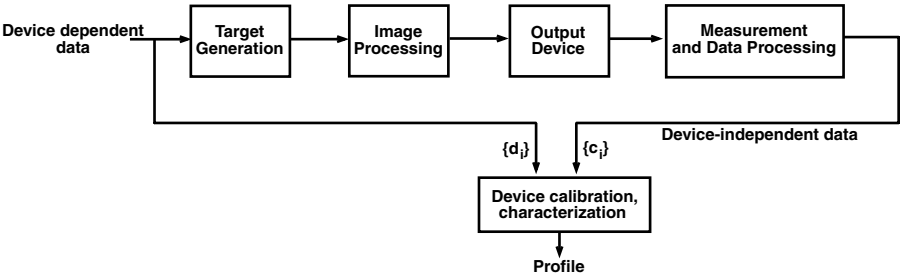


Figure 5.9 Output device characterization workflow.

independent color representations. This can be used to generate both calibration and characterization functions, in that order. For characterization, we once again derive a forward and an inverse function. The forward function describes the colorimetric response of the (calibrated) device to a certain device-dependent input. The inverse characterization function determines the device-dependent values that should be presented to a (calibrated) device to reproduce a certain colorimetric input.

As with input devices, the calibration and characterization should then be evaluated with an independent test target. The flow diagram for doing this is shown in Figure 5.10**.** The test target comprises a set of patches with known

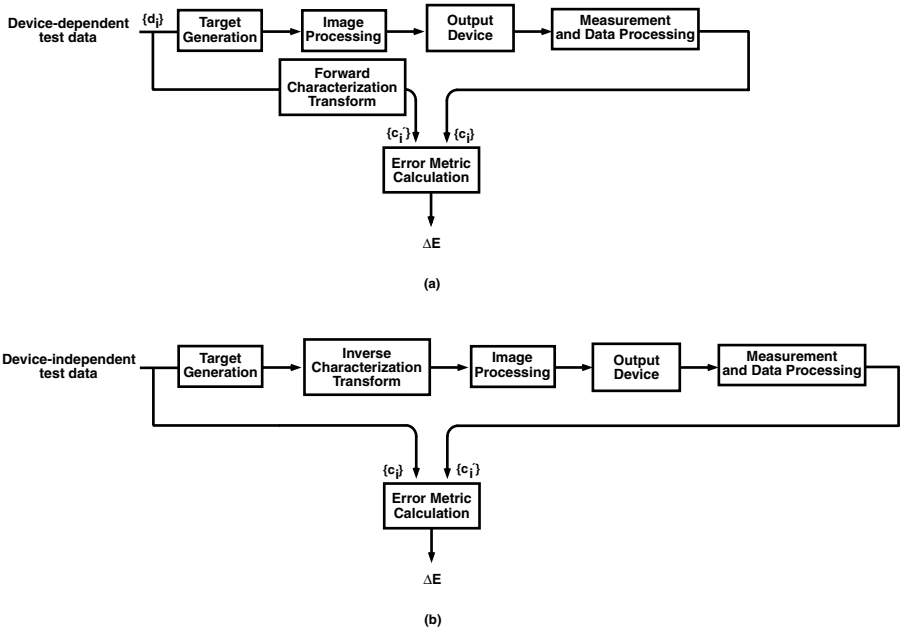


Figure 5.10 Testing of (a) forward and (b) inverse output device characterization.

device-independent coordinates. If calibration is being tested, this target is processed through the calibration functions and rendered to the device. If characterization is being evaluated, the target is processed through both the characterization and calibration function and rendered to the device. The resulting output is measured in device-independent coordinates and compared with the original target values. Once again, the comparison is to be carried out with an appropriate color difference formula such as ΔE^*_{ab} or ΔE^*_{94}.

An important component of the color characteristics of an output device is its color gamut, namely the volume of colors in three-dimensional colorimetric space that is physically achievable by the device. Of particular importance is the gamut surface, as this is used in gamut mapping algorithms. This information can easily be derived from the characterization process. Details of gamut surface calculation are provided in the chapter on gamut mapping.

5.3 Characterization targets and measurement techniques

The generation and measurement of color targets is an important component of device characterization. Hence, a separate section is devoted to this topic.

5.3.1 Color target design

The design of a color target involves several factors. First is the set of colorants and underlying medium of the target. In the case of input devices, the characterization target is created offline (i.e., it is not part of the characterization process) with colorants and media that are representative of what the device is likely to capture. For example, for scanner characterization, photographic and offset lithographic processes are commonly used to create targets on reflective or transmissive media. In the case of output devices, target generation is part of the characterization process and should be carried out using the same colorants and media that will be used for final color rendition.

The second factor is the choice of color patches. Typically, the patches are chosen to span the desired range of the colors to be captured (in the case of input devices) or rendered (in the case of output devices). Often, critical memory colors are included, such as flesh tones and neutrals. The optimal choice of patches is logically a function of the particular algorithm or model that will be used to generate the calibration or characterization function. Nevertheless, a few targets have been adopted as industry standards, and they accommodate a variety of characterization techniques. For input device characterization, these include the CGATS/ANSI IT8.7/1 and IT8.7/2 targets for transmission and reflection media respectively (http://webstore.ansi.org/ansidocstore); the Kodak photographic Q60 target, which is based on the IT8 standards and is made with Ektachrome dyes on Ektacolor paper (www.kodak.com); the GretagMacbeth ColorChecker chart (www.munsell.com); and ColorChecker DC version for digital cam-

eras (www.gretagmacbeth.com). For output device characterization, the common standard is the IT8.7/3 CMYK target (http://webstore.ansi.org/ansidocstore). The Q60 and IT8.7/3 targets are shown in Plates 5A and 5B.

A third factor is the spatial layout of the patches. If a device is known to exhibit spatial nonuniformity, it may be desirable to generate targets with the same set of color patches but rendered in different spatial layouts. The measurements from the multiple targets are then averaged to reduce the effect of the nonuniformity. In general, this approach is advised so as to reduce the overall effect of various imperfections and noise in the characterization process. In the case of input devices, target creation is often not within the practitioner's control; rather, the targets are supplied by a third-party vendor such as Eastman Kodak or Fuji Film. Generally, however, these vendors do use similar principles to generate reliable measurement data.

Another motivation for a specific spatial layout is visual inspection of the target. The Kodak Q60 target, for example, is designed with a gray ramp at the bottom and neutral colors all collected in one area. This allows for convenient visual inspection of these colors, to which we are more sensitive.

5.3.2 Color measurement techniques

5.3.2.1 Visual approaches

Most visual approaches rely on observers making color matching judgments. Typically, a varying stimulus produced by a given device is compared against a reference stimulus of known measurement. When a visual match is reported, this effectively provides a measurement for the varying stimulus and can be correlated with the device value that produced the stimulus. The major advantage of a visual approach is that it does not require expensive measurement instrumentation. Proponents also argue that the best color measurement device is the human visual system, because, after all, this is the basis for colorimetry. However, these approaches have their limitations. First, to achieve reliable results, the visual task must be easy to execute. This imposes severe limits on the number and nature of measurements that can be made. Second, observer-to-observer variation will produce measurements and a characterization that may not be satisfactory to all observers. Nevertheless, visual techniques are appealing in cases where the characterization can be described by a simple model and thus derived with a few simple measurements. The most common application of visual approaches is thus found in CRT characterization, discussed further in Section 5.8.3.

5.3.2.2 Instrument-based approaches

Color measurement instruments fall into two general categories, broadband and narrowband. A broadband measurement instrument reports up to three color signals obtained by optically processing the input light through broadband filters. Photometers are the simplest example, providing a measurement only of the luminance of a stimulus. Their primary use is in determin-

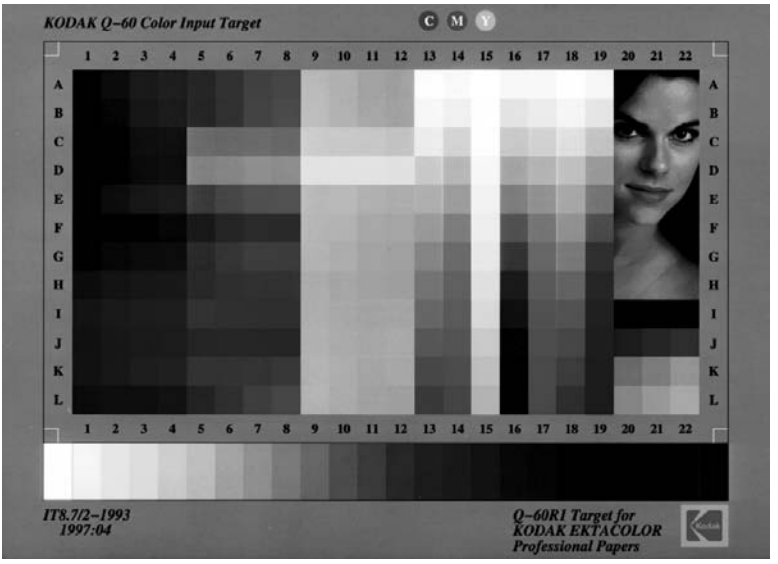


Figure 5A ***(See color insert following page 430)*** Q60 input characterization target.

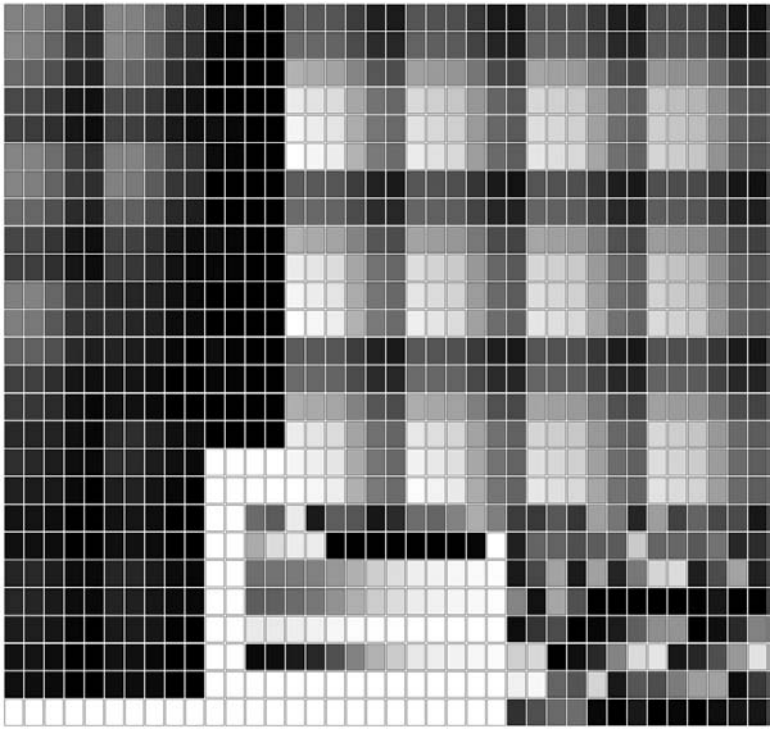


Figure 5B ***(See color insert)*** IT87/3 output characterization target.

ing the nonlinear calibration function of displays (discussed in Section 5.8). Densitometers are an example of broadband instruments that measure optical density of light filtered through red, green, and blue filters. Colorimeters are another example of broadband instruments that directly report tristimulus (XYZ) values and their derivatives such as CIELAB. In the narrowband category fall instruments that report spectral data of dimensionality significantly larger than three. Spectrophotometers and spectroradiometers are examples of narrowband instruments. These instruments typically record spectral reflectance and radiance, respectively, within the visible spectrum in increments ranging from 1 to 10 nm, resulting in 30 to 300 channels. They also have the ability to internally calculate and report tristimulus coordinates from the narrowband spectral data. Spectroradiometers can measure both emissive and reflective stimuli, while spectrophotometers can measure only reflective stimuli.

The main advantages of broadband instruments such as densitometers and colorimeters are that they are inexpensive and can read out data at very high rates. However, the resulting measurement is only an approximation of the true tristimulus signal, and the quality of this approximation varies widely, depending on the nature of the stimulus being measured. Accurate colorimetric measurement of arbitrary stimuli under arbitrary illumination and viewing conditions requires spectral measurements afforded by the more expensive narrowband instruments. Traditionally, the printing industry has satisfactorily relied on densitometers to make color measurements of prints made by offset ink. However, given the larger variety of colorants, printing technologies, and viewing conditions likely to be encountered in today's digital color imaging business, the use of spectral measurement instruments is strongly recommended for device characterization. Fortunately, the steadily declining cost of spectral instrumentation makes this a realistic prospect.

Instruments measuring reflective or transmissive samples possess an internal light source that illuminates the sample. Common choices for sources are tungsten-halogen bulbs as well as xenon and pulsed-xenon sources. An important consideration in reflective color measurement is the optical geometry used to illuminate the sample and capture the reflected light. A common choice is the 45/0 geometry, shown in Figure 5.11. (The two numbers are the angles with respect to the surface normal of the incident illumination and detector respectively.) This geometry is intended to minimize the effect of specular reflection and is also fairly representative of the conditions under which reflection prints are viewed. Another consideration is the measurement aperture, typically set between 3 and 5 mm. Another feature, usually offered at extra cost with the spectrophotometer, is a filter that blocks out ultraviolet (UV) light emanated by the internal source. The filter serves to reduce the amount of fluorescence in the prints that is caused by the UV light. Before using such a filter, however, it must be remembered that common viewing environments are illuminated by light sources (e.g., sunlight, fluorescent lamps) that also exhibit a significant amount of UV

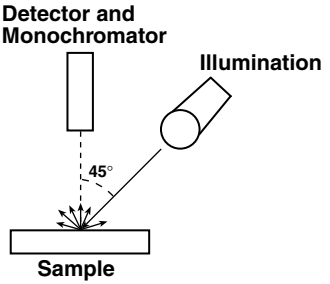


Figure 5.11 45/0 measurement geometry.

energy. Hence, blocking out UV energy may provide color measurements that are less germane to realistic viewing conditions.

For reflective targets, another important factor to consider is the color of the backing surface on which the target is placed for measurement. The two common options are black and white backing, both of which have advantages and disadvantages. A black backing will reduce the effect of show-through from the image on the backside of a duplex print. However, it will also expose variations in substrate transmittance, thus resulting in noisier measurements. A white backing, on the other hand, is not as effective at attenuating show-through; however, the resulting measurements are less noisy, because the effect of substrate variations is reduced. Generally, a white backing is recommended if the target is not duplex (which is typically the case.) Further details are provided by Rich.[9]

Color measurement instruments must themselves be calibrated to output reliable and repeatable data. Instrument calibration entails understanding and specifying many of the aforementioned parameters and, in some cases, needs to be carried out frequently. Details are provided by Zwinkel.[10]

Because color measurement can be a labor-intensive task, much has been done in the color management industry to automate this process. The Gretag Spectrolino™ product enables the target to be placed on a stage and automatically measured by the instrument. These measurements are then stored on a computer to be retrieved for deriving the characterization. In a similar vein, X-Rite Corporation has developed the DTP-41 scanning spectrophotometer. The target is placed within a slot in the "strip reader" and is automatically moved through the device as color measurements are made of each patch.

5.3.3 *Absolute and relative colorimetry*

An important concept that underlies device calibration and characterization is normalization of the measurement data by a reference white point. Recall from an earlier chapter that the computation of tristimulus XYZ values from spectral radiance data is given by

$$X = K \int_{\lambda \varepsilon V} S(\lambda)\bar{x}(\lambda)\partial\lambda, \quad Y = K \int_{\lambda \varepsilon V} S(\lambda)\bar{y}\partial\lambda, \quad Z = K \int_{\lambda \varepsilon V} S(\lambda)\bar{z}(\lambda)\partial\lambda \qquad (5.9)$$

where $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, $\bar{z}(\lambda)$ = color matching functions
V = set of visible wavelengths
K = a normalization constant

In absolute colorimetry, K is a constant, expressed in terms of the maximum efficacy of radiant power, equal to 683 lumens/W. In relative colorimetry, K is chosen such that $Y = 100$ for a chosen reference white point.

$$K = \frac{100}{\int_{\lambda \varepsilon V} S_w(\lambda)\bar{y}(\lambda)\partial\lambda} \qquad (5.10)$$

where $S_w(\lambda)$ = the spectral radiance of the reference white stimulus.

For reflective stimuli, radiance $S_w(\lambda)$ is a product of incident illumination $I(\lambda)$ and spectral reflectance $R_w(\lambda)$ of a white sample. The latter is usually chosen to be a perfect diffuse reflector (i.e., $R_w(\lambda) = 1$) so that $S_w(\lambda) = I(\lambda)$ in Equation 5.10.

There is an additional white-point normalization to be considered. The conversion from tristimulus values to appearance coordinates such as CIELAB or CIELUV requires the measurement of a reference white stimulus and an appropriate scaling of all tristimulus values by this white point. In the case of emissive display devices, the white point is the measurement of the light emanated by the display device when the driving RGB signals are at their maximal values (e.g., $D_R = D_G = D_B = 255$ for 8-bit input). In the case of reflective samples, the white point is obtained by measuring the light emanating from a reference white sample illuminated by a specified light source. If an ideal diffuse reflector is used as the white sample, we refer to the measurements as being in *media absolute colorimetric coordinates.* If a particular medium (e.g., paper) is used as the stimulus, we refer to the measurements as being in *media relative colorimetric coordinates.* Conversions between media absolute and relative colorimetry are achieved with a white-point normalization model such as the von Kries formula.

To get an intuitive understanding of the effect of media absolute vs. relative colorimetry, consider an example of scan-to-print reproduction of a color image. Suppose the image being scanned is a photograph whose medium typically exhibits a yellowish cast. This image is to be printed on a xerographic printer, which typically uses a paper with fluorescent whiteners and is thus lighter and bluer than the photographic medium. The image is scanned, processed through both scanner and printer characterization functions, and printed. If the characterizations are built using media absolute colorimetry, the yellowish cast of the photographic medium is

preserved in the xerographic reproduction. On the other hand, with media relative colorimetry, the "yellowish white" of the photographic medium maps directly to the "bluish white" of the xerographic medium under the premise that the human visual system adapts and perceives each medium as "white" when viewed in isolation. Arguments can be made for both modes, depending on the application. Side-by-side comparisons of original and reproduction may call for media absolute characterization. If the reproduction is to be viewed in isolation, it is probably preferable to exploit visual white-point adaptation and employ relative colorimetry. To this end, the ICC specification supports both media absolute and media relative modes in its characterization tables.

Finally, we remark that, while a wide variety of standard illuminants can be selected for deriving the device characterization function, the most common choices are CIE daylight illuminants D5000 (typically used for reflection prints) and D6500 (typically used for the white point of displays).

5.4 *Multidimensional data fitting and interpolation*

Another critical component underlying device characterization is multidimensional data fitting and interpolation. This topic is treated in general mathematical terms in this section. Application to specific devices will be discussed in ensuing sections.

Generally, the data samples generated by the characterization process in both device-dependent and device-independent spaces will constitute only a small subset of all possible digital values that could be encountered in either space. One reason for this is that the total number of possible samples in a color space is usually prohibitively large for direct measurement of the characterization function. As an example, for R, G, B signals represented with 8-bit precision, the total number of possible colors is 2^{24} = 16,777,216; clearly an unreasonable amount of data to be acquired manually. However, because the final characterization function will be used for transforming arbitrary image data, it needs to be defined for all possible inputs within some expected domain. To accomplish this, some form of data fitting or interpolation must be performed on the characterization samples. In model-based characterization, the underlying physical model serves to perform the fitting or interpolation for the forward characterization function. With empirical approaches, mathematical techniques may be used to perform data fitting or interpolation. Some of the common mathematical approaches are discussed in this section.

The fitting or interpolation concept can be formalized as follows. Define a set of T m-dimensional device-dependent color samples $\{\mathbf{d}_i\} \in R^m$, $i = 1, \ldots, T$ generated by the characterization process. Define the corresponding set of n-dimensional device-independent samples $\{\mathbf{c}_i\} \in R^n$, $i = 1, \ldots, T$. For the majority of characterization functions, $n = 3$, and $m = 3$ or 4. We will often refer to the pair $(\{\mathbf{d}_i\}, \{\mathbf{c}_i\})$ as the set of training samples. From this set, we wish to evaluate one or both of the following functions:

- f: $\boldsymbol{F} \in R^m \rightarrow R^n$, mapping device-dependent data within a domain $\boldsymbol{F}$ to device-independent color space
- g: $\boldsymbol{G} \in R^n \rightarrow R^m$, mapping device-independent data within a domain $\boldsymbol{G}$ to device-dependent color space

In interpolation schemes, the error of the functional approximation is identically zero at all the training samples, i.e., $f(\mathbf{d}_i) = \mathbf{c}_i$, and $g(\mathbf{c}_i) = \mathbf{d}_i$, $i = 1, \ldots, T$.

In fitting schemes, this condition need not hold. Rather, the fitting function is designed to minimize an error criterion between the training samples and the functional approximations at these samples. Formally,

$$f_{opt} = \arg\min_{f} E_1(|\mathbf{c}_i, f(\mathbf{d}_i)|_{i=1,\ldots,T}); \quad g_{opt} = \arg\min_{g} E_2(|\mathbf{d}_i, g(\mathbf{c}_i)|_{i=1,\ldots,T}) \tag{5.11}$$

where E_1 and E_2 are suitably chosen error criteria.

A common approach is to pick a parametric form for f (or g) and minimize the mean squared error metric, given by

$$E_1 = \frac{1}{T}\sum_{i=1}^{T} \| \boldsymbol{c}_i - f(\mathbf{d}_i) \|^2 \tag{5.12}$$

An analogous expression holds for E_2. The minimization is performed with respect to the parameters of the function f or g.

Unfortunately, most of the data fitting and interpolation approaches to be discussed shortly are too computationally expensive for the processing of large amounts of image pixel data in real time. The most common way to address this problem is to first evaluate the complex fitting or interpolation functions at a regular lattice of points in the input space and build a multidimensional lookup table (LUT). A fast interpolation technique such as trilinear or tetrahedral interpolation is then used to transform image data using this LUT. The subject of fast LUT interpolation on regular lattices is treated in a later chapter. Here, we will focus on the fitting and interpolation methods used to initially approximate the characterization function and build the LUT.

Often, it is necessary to evaluate the functions f and g within domains **F** and **G** that are outside of the volumes spanned by the training data $\{\mathbf{d}_i\}$ and $\{\mathbf{c}_i\}$. An example is shown in Figure 5.12 for printer characterization mapping CIELAB to CMY. A two-dimensional projection of CIELAB space is shown, with a set of training samples $\{\mathbf{c}_i\}$ indicated by "x." Device-dependent CMY values $\{\mathbf{d}_i\}$ are known at each of these points. The shaded area enclosed by these samples is the range of colors achievable by the printer, namely its color gamut. From these data, the inverse printer characterization function from CIELAB to CMY is to be evaluated at each of the lattice points lying on the three-dimensional lookup table grid (projected as a two-dimensional grid in

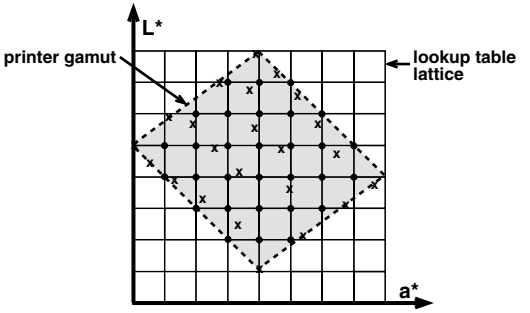


Figure 5.12 Multidimensional lattice in CIELAB, overlaying printer gamut.

Figure 5.12). Hence, the domain **G** in this case is the entire CIELAB cube. Observe that a fraction of these lattice points lie within the printer gamut (shown as black circles). Interpolation or data fitting of these points is usually well defined and mathematically robust, since a sufficient amount of training data is available in the vicinity of each lattice point. However, a substantial fraction of lattice points also lie outside the gamut, and there are no training samples in the vicinity of these points. One of two approaches can be used to determine the characterization function at these points.

1. Apply a preprocessing step that first maps all out-of-gamut colors to the gamut, then perform data fitting or interpolation to estimate output values.
2. Extrapolate the fitting or interpolation function to these out-of-gamut regions.

Some of the techniques described herewith allow for data extrapolation. The latter will invariably generate output data that lie outside the allowable range in the output space. Hence, some additional processing is needed to limit the data to this range. Often, a hard-limiting or clipping function is employed to each of the components of the output data.

Two additional comments are noteworthy. First, while the techniques described in this section focus on fitting and interpolation of multidimensional data, most of them apply in a straightforward manner to one-dimensional data typically encountered in device calibration. Linear and polynomial regression and splines are especially popular choices for fitting one-dimensional data. Lattice-based interpolation reduces trivially to piecewise linear interpolation, and it can be used when the data are well behaved and exhibit low noise. Secondly, the reader is strongly encouraged, where possible, to plot the raw data along with the fitting or interpolation function to obtain insight on both the characteristics of the data and the functional approximation. Often, data fitting involves a delicate balance between accurately approximating the function and smoothing out the noise. This balance

is difficult to achieve by examining only a single numerical error metric and is significantly aided by visualizing the entire dataset in combination with the fitting functions.

5.4.1 Linear least-squares regression

This very common data fitting approach is used widely in color imaging, particularly in device characterization and modeling. The problem is formulated as follows. Denote **d** and **c** to be the input and output color vectors, respectively, for a characterization function. Specifically, **d** is a $1 \times m$ vector, and **c** is a $1 \times n$ vector. We wish to approximate the characterization function by the linear relationship $\mathbf{c} = \mathbf{d} \cdot \mathbf{A}$.

The matrix **A** is of dimension $m \times n$ and is derived by minimizing the mean squared error of the linear fit to a set of training samples, $\{\mathbf{d}_i, \mathbf{c}_i\}$, $i = 1, \ldots, T$. Mathematically, the optimal **A** is given by

$$A_{opt} = \arg\min_{A} \left\{ \frac{1}{T} \sum_{i=1}^{T} \| \boldsymbol{c}_i - \mathbf{d}_i \mathbf{A} \|^2 \right\} \tag{5.13}$$

To continue the formulation, it is convenient to collect the samples $\{\mathbf{c}_i\}$ into a $T \times n$ matrix $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_T]$, and $\{\mathbf{d}_i\}$ into a $T \times m$ matrix $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_T]$. The linear relationship is given by $\mathbf{C} = \mathbf{D} \cdot \mathbf{A}$. The optimal **A** is given by $\mathbf{A} = \mathbf{D}\dagger\ \mathbf{C}$, where $\mathbf{D}\dagger$ is the generalized inverse (sometimes known as the Moore–Penrose pseudo-inverse) of **D**. In the case where $\mathbf{D}^t\mathbf{D}$ is invertible, the optimum **A** is given by

$$\mathbf{A} = (\mathbf{D}^t\mathbf{D})^{-1}\ \mathbf{D}^t\mathbf{C} \tag{5.14}$$

See Appendix 5.A for the derivation and numerical computation of this least-squares solution. It is important to understand the conditions for which the solution to Equation 5.14 exists. If $T < m$, we have an underdetermined system of equations with no unique solution. The mathematical consequence of this is that the matrix $\mathbf{D}^t\mathbf{D}$ is of insufficient rank and is thus not invertible. Thus, we need at least as many samples as the dimensionality of the input data. If $T = m$, we have an exact solution for **A** that results in the squared error metric being identically zero. If $T > m$ (the preferred case), Equation 5.14 provides a least-squares solution to an overdetermined system of equations. Note that linear regression affords a natural means of extrapolation for input data **d** lying outside the domain of the training samples. As mentioned earlier, some form of clipping will be needed to limit such extrapolated outputs to their allowable range.

5.4.2 Weighted least-squares regression

The standard least-squares regression can be extended to minimize a weighted error criterion,

$$\boldsymbol{A}_{opt} = \arg\min\left\{\frac{1}{T}\sum_{i=1}^{T} w_i \|\mathbf{c}_i - \mathbf{d}_i\mathbf{A}\|^2\right\} \tag{5.15}$$

where w_i = positive-valued weights that indicate the relative importance of the ith data point, $\{\mathbf{d}_i, \mathbf{c}_i\}$.

Adopting the notation in Section 5.4.1, a straightforward extension of Appendix 5.A results in the following optimum solution:

$$\mathbf{A} = (\mathbf{D}^t\mathbf{W}\mathbf{D})^{-1}\,\mathbf{D}^t\,\mathbf{W}\mathbf{C} \tag{5.16}$$

where $\mathbf{W}$ is a $T \times T$ diagonal matrix with diagonal entries w_i.

The resulting fit will be biased toward achieving greater accuracy at the more heavily weighted samples. This can be a useful feature in device characterization when, for example, we wish to assign greater importance to colors in certain regions of color space (e.g., neutrals, fleshtones, etc.). As another example, in spectral regression, it may be desirable to assign greater importance to certain wavelengths than others.

5.4.3 Polynomial regression

This is a special form of least-squares fitting wherein the characterization function is approximated by a polynomial. We will describe the formulation using, as an example, a scanner characterization mapping device RGB space to XYZ tristimulus space. The formulation is conceptually identical for input and output devices and for the forward and inverse functions.

The third-order polynomial approximation for a transformation from RGB to XYZ space is given by

$$X = \sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3} w_{X,l} R^i G^j B^k; \quad Y = \sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3} w_{Y,l} R^i G^j B^k;$$

$$Z = \sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3} w_{Z,l}\; R^i G^j B^k \tag{5.17}$$

where $w_{X,l}$, etc. = polynomial weights

l = a unique index for each combination of i, j, k

In practice, several of the terms in Equation 5.17 are eliminated (i.e., the weights w are set to zero) so as to control the number of degrees of freedom in the polynomial. Two common examples, a linear and third-order approximation,

are given below. For brevity, only the X term is defined; analogous definitions hold for Y and Z.

$$X = w_{X,0}R + w_{X,1}G + w_{X,2}B \tag{5.18a}$$

$$\begin{aligned} X = {} & w_{X,0} + w_{X,1}R + w_{X,2}G + w_{X,3}B + w_{X,4}RG + w_{X,5}GB \\ & + w_{X,6}RB + w_{X,7}R^2 + w_{X,8}G^2 + w_{X,9}B^2 + w_{X,10}RGB \end{aligned} \tag{5.18b}$$

In matrix-vector notation, Equation 5.17 can be written as

$$\begin{bmatrix} X & Y & Z \end{bmatrix} = \begin{bmatrix} 1 & R & G & \ldots & R^3 & G^3 & B^3 \end{bmatrix} \begin{bmatrix} w_{X,0} & w_{Y,0} & w_{Z,0} \\ w_{X,1} & w_{Y,1} & w_{z,3} \\ & \cdots & \\ w_{X,63} & w_{Y,63} & w_{Z,63} \end{bmatrix} \tag{5.19}$$

or more compactly,

$$\mathbf{c} = \mathbf{p} \cdot \mathbf{A} \tag{5.20}$$

where $\mathbf{c}$ = output XYZ vector
$\mathbf{p}$ = $1 \times Q$ vector of Q polynomial terms derived from the input RGB vector $\mathbf{d}$
$\mathbf{A}$ = $Q \times 3$ matrix of polynomial weights to be optimized

In the complete form, $Q = 64$. However, with the more common simplified approximations in Equation 5.18, this number is significantly smaller; i.e., $Q = 3$ and $Q = 11$, respectively.

Note from Equation 5.20 that the polynomial regression problem has been cast into a linear least-squares problem with suitable preprocessing of the input data $\mathbf{d}$ into the polynomial vector $\mathbf{p}$. The optimal $\mathbf{A}$ is now given by

$$\mathbf{A}_{opt} = \arg\min_{\mathrm{A}} \left\{ \frac{1}{T} \sum_{i=1}^{T} \left\| \mathbf{c}_i - \mathbf{p}_i \mathbf{A} \right\|^2 \right\} \tag{5.21}$$

Collecting the samples $\{\mathbf{c}_i\}$ into a $T \times 3$ matrix $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_T]$, and $\{\mathbf{p}_i\}$ into a $T \times Q$ matrix $\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_T]$, we have the relationship $\mathbf{C} = \mathbf{P} \cdot \mathbf{A}$. Following the formulation in Section 5.4.1, the optimal solution for $\mathbf{A}$ is given by

$$\mathbf{A} = (\mathbf{P}^t\mathbf{P})^{-1}\,\mathbf{P}^t\mathbf{C} \tag{5.22}$$

For the $Q \times Q$ matrix $(\mathbf{P}^t\mathbf{P})$ to be invertible, we now require that $T \geq Q$.

Polynomial regression can be summarized as follows:

1. Select a set of T training samples, where $T > Q$, the number of terms in the polynomial approximation. It is recommended that the samples adequately span the input color space.
2. Use the assumed polynomial model to generate the polynomial terms $\mathbf{p}_i$ from the input data $\mathbf{d}_i$. Collect $\mathbf{c}_i$ and $\mathbf{p}_i$ into matrices $\mathbf{C}$ and $\mathbf{P}$, respectively.
3. Use Equation (5.22 to derive the optimal $\mathbf{A}$.
4. For a given input color $\mathbf{d}$, use the same polynomial model to generate the polynomial terms $\mathbf{p}$.
5. Use Equation 5.20 to compute the output color $\mathbf{c}$.

Figure 5.13 is a graphical one-dimensional example of different polynomial approximations to a set of training samples. The straight line is a linear fit (Q = 3) and is clearly inadequate for the given data. The solid curve is a second-order polynomial function (Q = 7) and offers a much superior fit. The dash–dot curve closely following the solid curve is a third-order polynomial approximation (Q = 11). Clearly, this offers no significant advantage over the second-order polynomial. In general, we recommend using the smallest number of polynomial terms that adequately fits the curvature of the function while still smoothing out the noise. This choice is dependent on the particular device characteristics and is obtained by experimentation, intuition, and experience. Finally, it is noted that polynomial regression affords a natural means of extrapolation for input data lying outside the domain of the training samples.

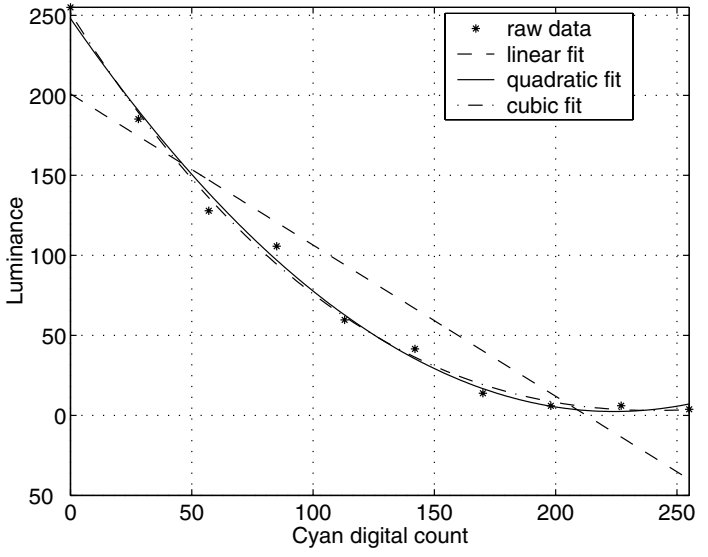


Figure 5.13 One-dimensional example of different polynomial approximations.

5.4.4 Distance-weighted techniques

The previous section described the use of a global polynomial function that results in the best overall fit to the training samples. In this section, we describe a class of techniques that also employ simple parametric functions; however, the parameters vary across color space to best fit the local characteristics of the training samples.

5.4.4.1 Shepard's interpolation

This is a technique that can be applied to cases in which the input and output spaces of the characterization function are of the same dimensionality. First, a crude approximation of the characterization function is defined: $\hat{c} = f_{approx}(\mathbf{d})$. The main purpose of $f_{approx}()$ is to bring the input data into the orientation of the output color space. (By "orientation," it is meant that all RGB spaces are of the same orientation, as are all luminance–chrominance spaces, etc.) If both color spaces are already of the same orientation, e.g., printer RGB and sRGB, we can simply let $f_{approx}()$ be an identity function so that $\hat{c} = \mathbf{d}$. If, for example, the input and output spaces are scanner RGB and CIELAB, an analytic transformation from any colorimetric RGB (e.g., sRGB) to CIELAB could serve as the crude approximation.

Next, given the training samples $\{\mathbf{d}_i\}$ and $\{\mathbf{c}_i\}$ in the input and output space, respectively, we define error vectors between the crude approximation and true output values of these samples: $e_i = c_i - \hat{c}_i, = 1, \ldots, T$. Shepard's interpolation for an arbitrary input color vector $\mathbf{d}$ is then given by[11]

$$\mathbf{c} = \hat{\mathbf{c}} + K_w \sum_{i=1}^{T} w(\mathbf{d} - \mathbf{d}_i)\mathbf{e}_i \tag{5.23}$$

where $w()$ = weights

K_w = a normalizing factor that ensures that these weights sum to unity as follows:

$$K_w = \frac{1}{\sum_{i=1}^{T} w(\mathbf{d} - \mathbf{d}_i)} \tag{5.24}$$

The second term in Equation 5.23 is a correction for the residual error between $\mathbf{c}$ and $\hat{\mathrm{c}}$, and it is given by a weighted average of the error vectors $\mathbf{e}_i$ at the training samples. The weighting function $w()$ is chosen to be inversely proportional to the Euclidean distance between $\mathbf{d}$ and $\mathbf{d}_i$ so that training samples that are nearer the input point exhibit a stronger influence than those that are further away. There are numerous candidates for $w()$. One form that has been successfully used for printer and scanner characterization is given by[12]

$$w(\mathbf{d} - \mathbf{d}_i) = \frac{1}{\|\mathbf{d} - \mathbf{d}_i\|^p + \varepsilon} \tag{5.25}$$

where $\|\mathbf{d} - \mathbf{d}_i\|$ denotes Euclidean distance between vectors $\mathbf{d}$ and $\mathbf{d}_i$, and ρ and ε are parameters that dictate the relative influence of the training samples as a function of their distance from the input point. As ρ increases, the influence of a training sample decays more rapidly as a function of its distance from the input point. As ε increases, the weights become less sensitive to distance, and the approach migrates from a local to a global approximation.

Note that, in the special case where ε = 0, the function in Equation 5.25 has a singularity at $\mathbf{d} = \mathbf{d}_i$. This can be accommodated by adding a special condition to Equation 5.23.

$$\mathbf{c} = \begin{cases} \hat{\mathbf{c}} + K_w \sum_{i=1}^{T} w(\mathbf{d} - \mathbf{d}_i)\mathbf{e}_i & if(\|\mathbf{d} - \mathbf{d}_i\| \geq t) \\ \mathbf{c}_i & if\|\mathbf{d} - \mathbf{d}_i\| < t \end{cases} \tag{5.26}$$

where $w()$ = given by Equation 5.25 with ε = 0
t = a suitably chosen distance threshold that avoids the singularity at $\mathbf{d} = \mathbf{d}_i$

Other choices of $w()$ include the Gaussian and exponential functions.[11] Note that, depending on how the weights are chosen, Shepard's algorithm can be used for both data fitting (i.e., Equation 5.23 and Equation 5.25 with ε > 0), and data interpolation, wherein the characterization function coincides exactly at the training samples (i.e., Equation 5.26). Note also that this technique allows for data extrapolation. As one moves farther away from the volume spanned by the training samples, the distances $\|\mathbf{d} - \mathbf{d}_i\|$ and hence the weights $w()$ approach a constant. In the limit, the overall error correction in Equation 5.23 is an unweighted average of the error vectors $\mathbf{e}_i$.

5.4.4.2 Local linear regression

In this approach, the form of the characterization function that maps input colors $\mathbf{d}$ to output colors $\mathbf{c}$ is given by

$$\mathbf{c} = \mathbf{d} \cdot \mathbf{A}_d \tag{5.27}$$

This looks very similar to the standard linear transformation, the important difference being that the matrix $\mathbf{A}_d$ now varies as a function of the input color $\mathbf{d}$ (hence the term *local linear regression)*. The optimal $\mathbf{A}_d$ is obtained by a distance-weighted least-squares regression,

$$\mathbf{A}_d^{opt} = \arg\min\left\{\frac{1}{T}\sum_{i=1}^{T}\|\mathbf{c}_i - \mathbf{d}_i\mathbf{A}_d\|^2 w(\mathbf{d}-\mathbf{d}_i)\right\} \tag{5.28}$$

As with Shepard's interpolation, the weighting function $w()$ is inversely proportional to the Euclidean distance $\|\boldsymbol{d} - \boldsymbol{d}_i\|$, so training samples $\mathbf{d}_i$ that are farther away from the input point $\mathbf{d}$ are assigned a smaller weight than nearby points. A form such as Equation 5.25 may be used.[12] The solution is given by Equation 5.16 in Section 4.2, where the weights $w(\mathbf{d} - \mathbf{d}_i)$ constitute the diagonal terms of $\mathbf{W}$. Note that because $w()$ is a function of the input vector $\mathbf{d}$, Equation 5.16 must be recalculated for every input vector $\mathbf{d}$. Hence, this is a computationally intensive algorithm. Fortunately, as noted earlier, this type of data fitting is not applied to image pixels in real time. Instead, it is used offline to create a multidimensional lookup table.

Figure 5.14 is a one-dimensional example of the locally linear transform using the inverse-distance weighting function, Equation 5.25. As with Shepard's interpolation, ρ and ε affect the relative influence of the training samples as a function of distance. The plots in Figure 5.14 were generated with ρ = 4 and compare two values of ε. For ε = 0.001, the function closely follows the data. As ε increases to 0.01, the fit averages the fine detail while preserving the gross curvature. In the limit as ε increases, $w()$ in Equation 5.25 approaches a constant, the technique approaches global linear regression, and the fit approaches a straight line. Similar trends hold for ρ. These param-

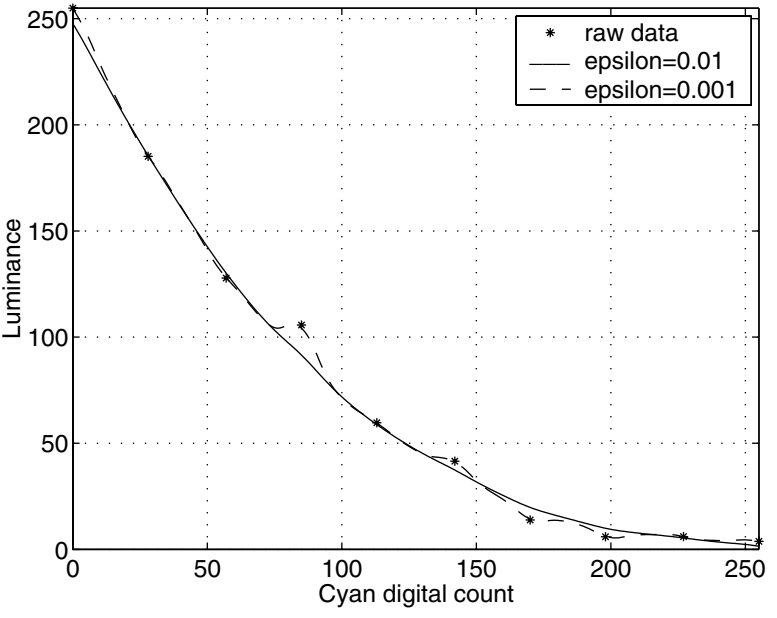


Figure 5.14 Local linear regression for different values of ε.

eters thus offer direct control on the amount of curvature and smoothing that occurs in the data fitting process and should be chosen based on *a priori* knowledge about the device and noise characteristics.

As with Shepard's algorithm, this approach also allows for data extrapolation. As the input point moves farther away from the volume spanned by the training samples, the weights $w()$ approach a constant, and we are again in the regime of global linear extrapolation.

5.4.5 Lattice-based interpolation

In this class of techniques, the training samples are assumed to lie on a regular lattice in either the input or output space of the characterization function. Define l_i to be a set of real-valued levels along the ith color dimension. A regular lattice L^m in m-dimensional color space is defined as the set of all points $\mathbf{x} = [x_1, \ldots, x_m]^t$ whose ith component x_i belongs to the set l_i. Mathematically, the lattice can be expressed as

$$L^m = \{x \in R^m \mid x_i \in l_i, i = 1,\ldots,m\} \text{ or, equivalently, } L^m = \prod_{i=1}^{m} l_i \qquad (5.29)$$

where the second expression is a Cartesian product. If s_i is the number of levels in l_i, the size of the lattice is the product $s_1 \times s_2 \times \ldots \times s_m$. Commonly, all the l_i are identical sets of size s, resulting in a lattice of size s^m.

In one dimension, a lattice is simply a set of levels $\{x_j\}$ in the input space. Associated with these levels are values $\{y_j\}$ in the output space. Evaluation of the one-dimensional function for an intermediate value of x is then performed by finding the interval $[x_j, x_{j+1}]$ that encloses x and performing piecewise interpolation using either linear or nonlinear functions. If sufficient samples exist and exhibit low noise, linear interpolation can be used as follows:

$$y = y_j + \left(\frac{x - x_j}{x_{j+1} - x_j}\right)(y_{j+1} - y_j) \qquad (5.30)$$

If only a sparse sampling is available, nonlinear functions such as splines may be a better choice (see Section 5.4.8).

Let us turn to the more interesting multidimensional case. A three-dimensional lattice in CMY space is shown in Figure 5.15, along with the corresponding lattice in CIELAB space. The lines indicate the levels l_i along each dimension, and the intersections of these lines are the lattice points. The lattice size in this example is $5 \times 5 \times 5 = 125$. A lattice partitions a color space into a set of smaller subvolumes. The characterization transform is executed in two steps: (1) Locate the subvolume to which an input color belongs, and (2) perform some form of interpolation, effectively a distance-weighted average, among the neighboring lattice points. By definition, the characterization function will coincide with the training samples at the lattice points.

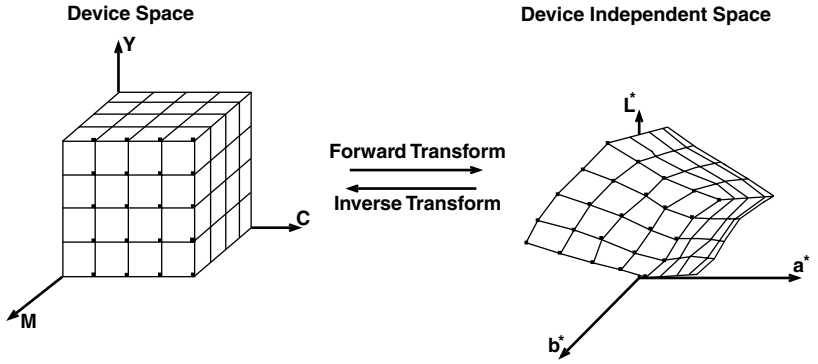


Figure 5.15 Three-dimensional lattice in CMY and CIELAB space.

Note from Figure 5.15 that, while the lattice is regular in one space, it need not be regular in the other space. In the case of the forward characterization function for an output device, the regular lattice exists in the input domain of the function. Efficient interpolation techniques exist for regular lattices, including trilinear, tetrahedral, prism, and pyramidal interpolation. These are described in detail in Chapter 11 and thus will not be discussed here. The more challenging case is evaluation of the inverse transform, whereby the lattice that partitions the input domain of the function is irregular. We will describe a solution to this problem known as tetrahedral inversion.[13] Let us assume that the dimensionality of both input and output color spaces are equal and assume, without less of generality, that the data are three-dimensional. A regular lattice in three-dimensional space provides a partitioning into a set of sub-cubes. Each sub-cube can be further partitioned into several tetrahedra, as shown in Figure 5.16. A tetrahedron is a volume bounded by four vertices and four planar surfaces. There are several ways to split a cube into tetrahedra, the most common form being a partitioning

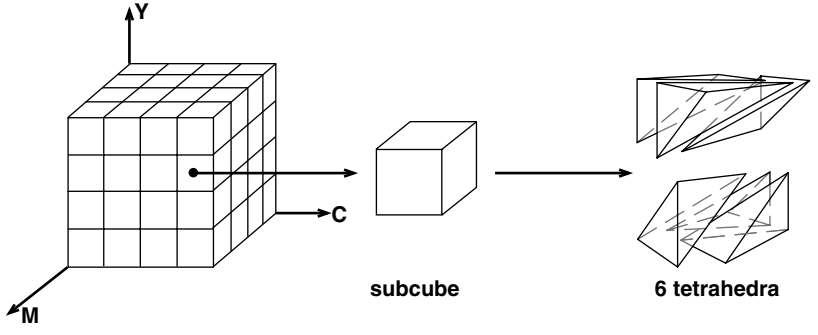


Figure 5.16 Partitioning of color space into cubes, further subdivided into tetrahedra.

into six tetrahedra that share a common diagonal of the cube. An association is now established between each quadruplet of vertices that constitute a tetrahedron on the regular lattice in device space and the corresponding quadruplet of vertices on the irregular lattice in device-independent space, as shown in Figure 5.17. The inverse characterization function $g()$ is then modeled as one that maps each tetrahedral volume in device-independent space to a corresponding tetrahedral volume in device space.

Specifically, referring to Figure 5.17, let $\{\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_4\}$ be four vertices of a tetrahedron $\mathbf{T}_d$ in device space, and $\{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4\}$ be the corresponding vertices forming a tetrahedron $\mathbf{T}_c$ in device-independent space. Here, $\mathbf{d}_i$ and $\mathbf{c}_i$ are 3×1 vectors. Given a point $\mathbf{c}$ lying within $\mathbf{T}_c$, the corresponding point $\mathbf{d}$ in $\mathbf{T}_d$ is given by

$$\mathbf{d} = g(\mathbf{c}) = \mathbf{A}_\mathbf{d} \cdot \mathbf{A}_\mathbf{c}^{-1} \cdot (\mathbf{c} - \mathbf{c}_1) + \mathbf{d}_1 \tag{5.31}$$

where $\mathbf{A}_\mathbf{d}$ and $\mathbf{A}_\mathbf{c}$ are 3×3 matrices given by

$$\mathbf{A}_\mathbf{d} = [\mathbf{d}_2 - \mathbf{d}_1 \;\; \mathbf{d}_3 - \mathbf{d}_1 \;\; \mathbf{d}_4 - \mathbf{d}_1]; \;\; \mathbf{A}_\mathbf{c} = [\mathbf{c}_2 - \mathbf{c}_1 \;\; \mathbf{c}_3 - \mathbf{c}_1 \;\; \mathbf{c}_4 - \mathbf{c}_1] \tag{5.32}$$

Equation 5.31 tells us that $g()$ is being modeled as a piecewise affine function. It can be shown that $\mathbf{c}$ is included within a tetrahedron $\mathbf{T}_c$ if all the elements of the vector $\mathbf{A}_\mathbf{c}^{-1}(\mathbf{c} - \mathbf{c}_1)$ are nonnegative and their sum lies between 0 and 1.[13]

Tetrahedral inversion may be summarized as follows:

- Partition the regular lattice of training samples into a set of tetrahedra.
- Establish a correspondence between tetrahedra on the regular lattice in the one space and tetrahedra on the possibly irregular lattice in the other space.
- Given an input point $\mathbf{c}$, find the tetrahedron $\mathbf{T}_\mathbf{c}$ to which the point belongs, using the aforementioned membership test.

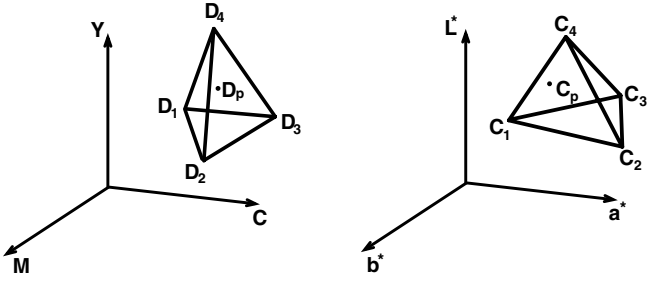


Figure 5.17 Tetrahedral mapping from device CMY space to colorimetric CIELAB space.

- Use Equations 5.31 and 5.32 to evaluate the characterization function $\mathbf{d} = g(\mathbf{c})$.

Because tetrahedral inversion requires membership in a tetrahedron, it does not allow extrapolation to points **c** that lie outside the lattice defined by the training samples. Hence, such points must first be mapped to the lattice volume before carrying out the inversion algorithm. Also, it is worth noting that tetrahedral interpolation on a regular lattice can be implemented with a highly simplified form of Equation 5.31. These equations will be included in the chapter on efficient color transformations.

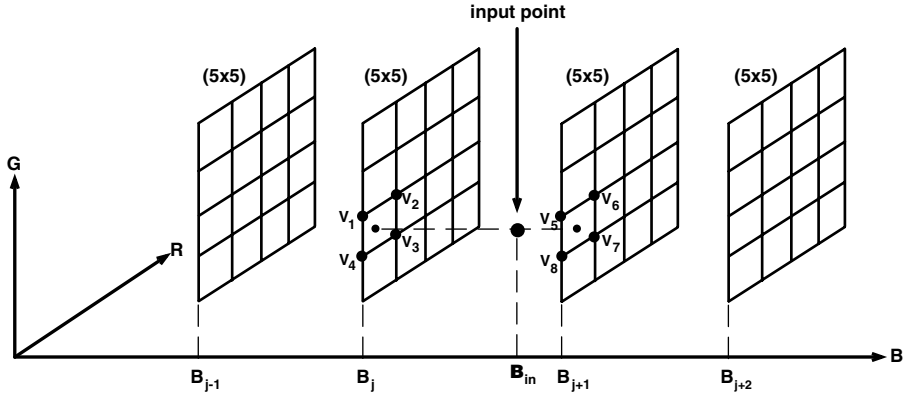
In the context of deriving a characterization function, regular lattices of training data can occur only for the case of output devices, as the patches in the color target can be designed to lie on a regular lattice in device space. With input device characterization, neither the captured device values nor the measured device-independent values of the color target can be guaranteed to lie on a regular lattice.

5.4.6 Sequential interpolation

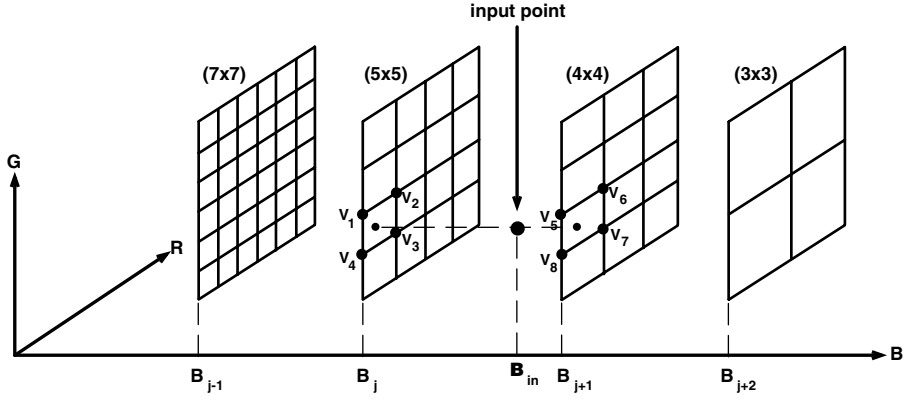
A primary advantage of a regular lattice is that it facilitates simple interpolation techniques. However, it limits the freedom in the placement of control points in multidimensional color space. Referring to Figure 5.12, one would expect considerable curvature of the characterization function in certain regions within the device gamut, while large regions outside the gamut would never be used for interpolation calculations. It would be desirable, therefore, to finely sample regions within the gamut, and coarsely sample regions far away from the gamut. As shown in the figure, the regular lattice does not permit this. A simple extension of regular lattice interpolation, which we term *sequential interpolation (SI)*, brings additional flexibility at a modest increase in computational cost.

In general terms, SI can be thought of as a two-stage interpolation process. Consider a decomposition of the space R^m into two subspaces of dimensions p and q, i.e., $R^m = R^p \times R^q$, $m = p + q$. The m-dimensional lattice L^m can also be decomposed into two sub-lattices L^p and L^q. Let s be the size of L^q. We can think of L^m as being a family of s p-dimensional lattices. In a conventional regular lattice each p-dimensional lattice is identical, and we have $L^m = L^p \times L^q$. In sequential interpolation, we let the p-dimensional lattice structure vary as a function of the remaining q dimensions.

To crystallize this concept, consider the three-dimensional lattice in Figure 5.18 used to implement a characterization function from device RGB to CIELAB. This lattice can be conceived as a family of two-dimensional RG lattices, corresponding to different levels of the third-dimension B. In Figure 5.18a, the RG lattices are identical as a function of B, which corresponds to a regular lattice in RGB space. In this case, interpolation of an input RGB point is accomplished by selecting a subset of the eight vertices $V_1, \ldots, V_8$ that enclose the point and performing a weighted average of the output values at



(a)



(b)

Figure 5.18 Comparison of (a) conventional and (b) sequential interpolation lattices.

these vertices. In Figure 5.18b, a sequential structure is shown where the RG lattice structure is allowed to change as a function of B. The interpolation calculation is accomplished by first projecting an input RGB point onto the B dimension and selecting the neighboring levels B_j and B_{j+1}. These correspond to two lattices in RG space. The input RGB point is then projected onto RG space, and two-dimensional interpolation is performed within each of these lattices, yielding two output colors $\mathbf{c}_j$, $\mathbf{c}_{j+1}$. Finally, one-dimensional interpolation is performed in the B dimension to produce the final output color. In this example, SI would be advantageous if the characterization function is known to exhibit different degrees of curvature for different values of B. If, for example, the function curvature is high for small values of B, SI permits a finer lattice sampling in these regions (as shown in Figure 5.18). Thus, with more efficient node placement, SI enables a given level of accuracy to be achieved with fewer lattice nodes than can be achieved with a regular lattice.

Figure 5.19 is a flow diagram showing the general case of SI in m-dimensions. Application of SI to CMYK printer characterization will be described in Section 5.10.3. Another special case of SI is sequential linear interpolation (SLI).[14] In SLI, we decompose the m-dimensional space into $(m - 1)$ dimensional and one-dimensional subspaces, then decompose the former into $(m - 2)$ and one-dimensional subspaces, and so on until we have a sequence of one-dimensional interpolations. SLI is described in more detail in Chapter 11.

5.4.7 *Neural networks*

Neural networks have taken inspiration from natural computational processes such as the brains and nervous systems of humans and animals. This class of techniques has received much attention in color imaging in recent years. In this section, we briefly describe the use of neural nets in device characterization, referring the reader to Masters[15] for excellent overviews, algorithms, and further reading on the subject.

A neural network is an interconnected assembly of simple processing units called *neurons* whose functionality is loosely based on the biological neuron. The processing ability of the network is stored in the inter-neuron connection strengths, or weights, obtained by a process of adaptation to, or learning from, a set of training patterns. In the most common configuration, the neurons are arranged into two or more layers, with inputs to neurons in a given layer depending exclusively on the outputs of neurons in previous layers. An example of such a multilayer feed-forward neural network is shown in Figure 5.20. This network has three inputs, three outputs, and one hidden layer of four neurons. The inputs are obtained from an external source (e.g., in our application, color data from the characterization process), and the outputs are the neural network's approximation of the response to these inputs. Let $s_i^{(L)}$ be the ith neuron in the Lth layer, $i = 1, \ldots, N_L$. The output from unit $s_i^{(L)}$ is given by

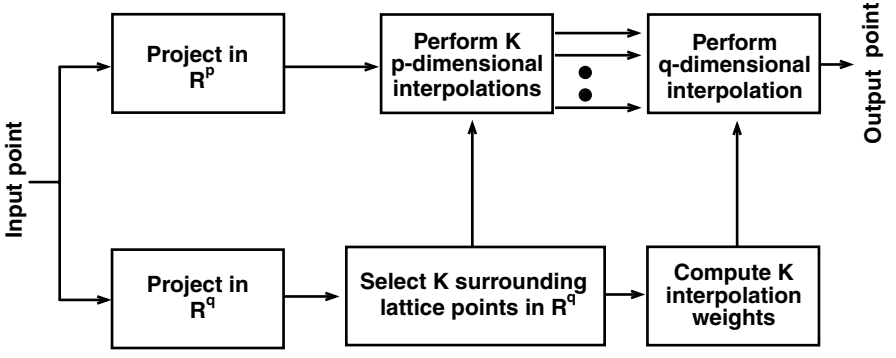


Figure 5.19 Block diagram of sequential interpolation.

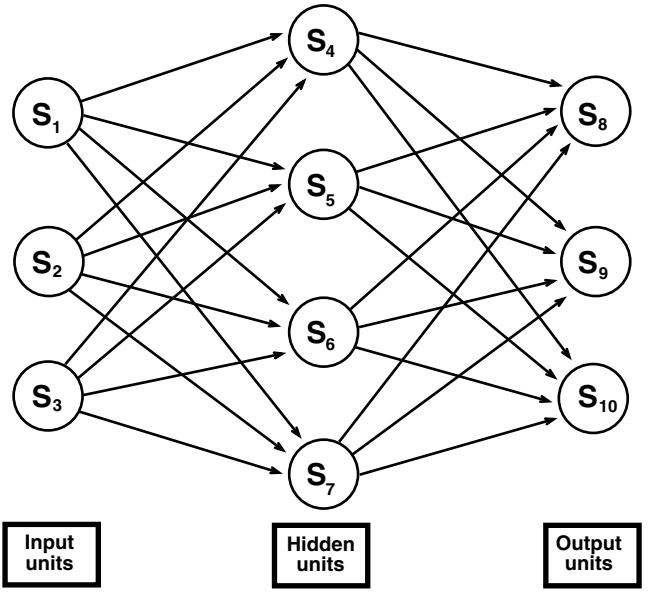


Figure 5.20 Three-layer (3–4–3) neural network.

$$s_i^{(L)} = h\left(\sum_{j=1}^{N_{L-1}} w_{ij} s_j^{(L-1)}\right) \tag{5.33}$$

where w_{ij} = a synaptic weight that determines the relative strength of the contribution of neuron $s_j^{(L-1)}$ to neuron $s_i^{(L)}$

function $h()$ = a nonlinear function, such as a step function or sigmoidal (S-shaped) function

Examples of sigmoidal functions are the logistic function, cumulative Gaussian, and hyperbolic tangent.[15] Depending on the particular architecture being implemented, constraints such as monotonicity and differentiability are often imposed on $h()$. The functionality of the overall neural net is determined by the number of layers and number of neurons per layer, the interconnecting links, the choice of $h()$, and the weights w_{ij}. Note from Equation 5.33 that each layer feeds only to the immediately following layer; this is the most typical configuration.

A popular method for neural network optimization is back-propagation, where all parameters except the synaptic weights w_{ij} are chosen beforehand, preferably based on some *a priori* knowledge about the nature of the function being approximated. The w_{ij} are then derived during a learning process in which a set of training samples in both input and output spaces is presented

to the network. An error metric such as the mean squared error in Equation 5.12 is minimized at the training samples with respect to w_{ij}. Because the overall neural network is a complex nonlinear function of w_{ij}, iterative error minimization approaches are called for. An example is the gradient descent algorithm, where a weight $w_{ij}^{(k)}$ at iteration k is given by

$$w_{ij}^{(k)} = w_{ij}^{(k-1)} - R\left(\frac{\delta E}{\delta w_{ij}}\right) \tag{5.34}$$

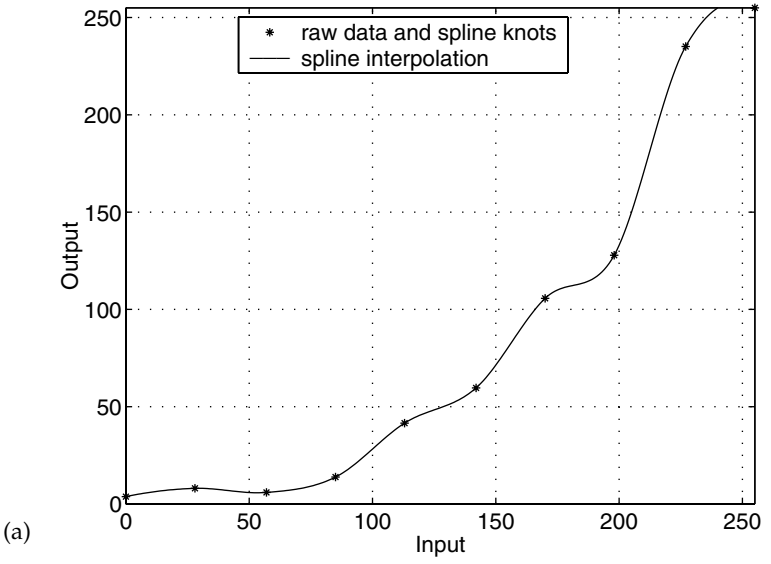
Here, E is the error metric being minimized, and R is a parameter known as the learning rate. The iteration continues until some convergence criterion is met with respect to the magnitude or the rate of change of E. The parameter R dictates the speed and stability of convergence. A major shortcoming of the gradient descent algorithm is that convergence is often unacceptably slow. An alternative search technique favored for significantly faster convergence is the conjugate gradient algorithm. As with all iterative algorithms, rate of convergence also depends on the choice of initial estimates, i.e., $w_{ij}^{(0)}$. Linear regression can be used to generate good initial estimates. Details are given in the book by Masters.[15]

The application to color characterization should be evident. A neural network can be used to approximate either the forward or inverse characterization functions. The training samples are the device-dependent and device-independent colors $\{\mathbf{c}_i, \mathbf{d}_i\}$ obtained in the characterization process. After the neural net is trained, arbitrary color inputs can now be processed through the network. The architecture of the network is chosen based on the expected complexity of the characterization function. As with polynomials, increased complexity can result in a better fit up to a certain point, beyond which the network will begin to track the noise in the data.

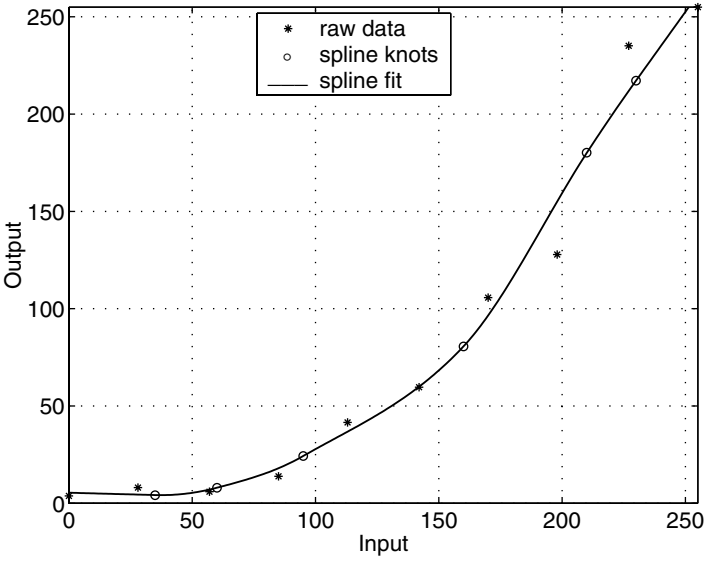
Typically, the iterative training can be a highly computationally intensive process. Fortunately, this is not a major concern, as this step is carried out offline. Neural networks are also usually too computationally intensive for real-time processing of image pixels. They can, however, be approximated by multidimensional LUTs, which are more computationally efficient.

5.4.8 Spline fitting

Spline interpolation constitutes a rich and flexible framework for approximating free-form shapes. One-dimensional splines can be used very effectively for the calibration step, whereas the multidimensional versions are applicable for characterization. The most common spline functions comprise a set of piecewise polynomial functions defined over a partition of segments in the input space, as shown for the one-dimensional case in Figure 5.21. The behavior of the spline is dictated by control points, known as *knots*, at the segment boundaries. The parameters of the polynomials are determined so that the function passes through all the knots while maintaining certain degrees of continuity across the segment boundaries.



(a)



(b)

Figure 5.21 Spline function used for (a) interpolation and (b) fitting.

Splines can be used for both interpolation and fitting. In the case of interpolation, shown in Figure 5.21a, the knots coincide with the data points. This approach is desirable when very few accurate data points are available. In the case of fitting, shown in Figure 5.21b, the control points do not necessarily coincide with the data and are actually free parameters chosen to minimize an error criterion between the data points and the spline fit. This approach is preferred when ample data is available but expected to be noisy

and therefore requiring some smoothing. The number and location of the knots used for spline fitting are critical. Too few knots could result in an excessively "stiff" spline that is unable to follow the curvature of the function, but too many knots could result in overshoots that follow the noise. A general guideline is to use fewer knots than data points and to space them approximately uniformly except in regions known to exhibit high curvature, where a denser sampling of knots can be used. As advised earlier, it is highly instructive to first plot and visualize the raw data so as to choose the knots appropriately.

The major advantage of splines over straightforward polynomial approximation is that the complexity of a spline can be tailored to suit the local characteristics of the function. Equivalently, a local change in a calibration or characterization function can be accurately approximated with a change in one local segment of a spline curve. Piecewise cubic and B-splines are popular choices for data fitting applications. Figure 5.22 is a comparison of cubic spline interpolation with the third-order polynomial approximation using the same data as in Figure 5.13. Clearly, the spline is capable of following the data more closely.

Space constraints do not permit a detailed treatment of splines in this chapter. The reader is referred to the book by Farin[16] for a comprehensive tutorial on the subject. C programs for cubic spline interpolation can be found in *Numerical Recipes in C*.[17] Users of Matlab can find an extensive set of spline functions in the spline toolbox (go to www.mathworks.com for details). As with other data-fitting techniques, the most suitable choice of spline function requires knowledge of the nature of the characterization data.

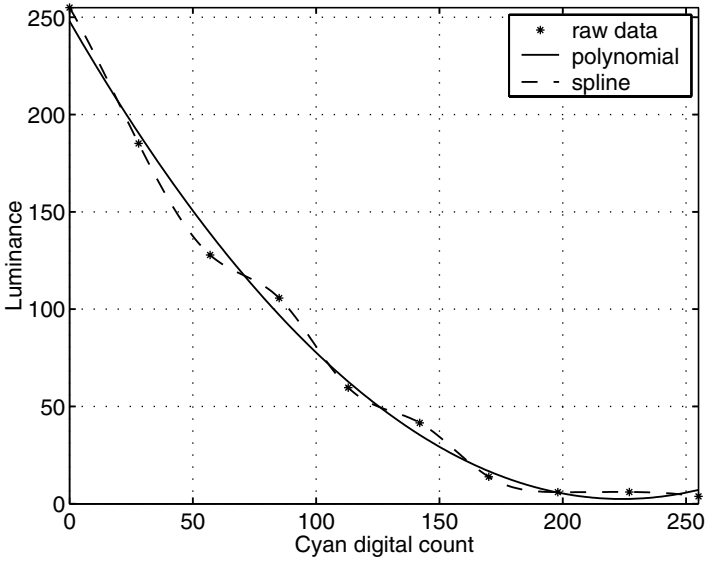


Figure 5.22 Comparison of spline and polynomial fitting.

5.5 Metrics for evaluating device characterization

Many of the mathematical techniques described in the previous section minimize quantitative error metrics. The resulting error from the fitting or interpolation is one indicator of the overall accuracy of characterization. However, this information is not sufficient, for several reasons:

1. The error is available only for the training samples.
2. The error is not always calculated in a visually meaningful color space.
3. Noise and other imperfections that can occur with multiple uses of the device are implicitly ignored.

To address the first concern, the notion of evaluating the characterization with independent test targets was introduced in Section 5.2. To address the second issue, evaluation of errors with visually relevant metrics is strongly recommended. While color difference formulae are described in detail in an earlier chapter, two of them, ΔE^*_{ab} and ΔE^*_{94} are restated here, as they are used extensively in this chapter. Given two CIELAB colors, and their component-wise differences, ΔL^*, Δa^*, Δb^* (equivalently, ΔL^*, ΔC^*, ΔH^*), the ΔE^*_{ab} color difference formula is simply the Euclidean distance between the two points in CIELAB space,

$$\Delta E^*_{ab} = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2} = \sqrt{(\Delta L^*)^2 + (\Delta C^*)^2 + (\Delta H^*)^2} \qquad (5.35)$$

It is important to bear in mind that ΔH^* is not a component-wise hue difference but rather is given by

$$\Delta H^* = \sqrt{(\Delta E^*_{ab})^2 - (\Delta L^*)^2 - (\Delta C^*)^2} \qquad (5.36)$$

The ΔE^*_{94} formula is an extension of ΔE^*_{ab} that applies different weights to the various components as follows:

$$\Delta E^*_{94} = \sqrt{\left(\frac{\Delta L^*}{k_L S_L}\right)^2 + \left(\frac{\Delta C^*}{k_C S_C}\right)^2 + \left(\frac{\Delta H^*}{k_H S_H}\right)^2} \qquad (5.37)$$

where $S_L = 1$
$S_C = 1 + 0.045\ C^*$
$S_H = 1 + 0.015\ C^*$

The parameters k_L, k_C, and k_H account for the effect of viewing conditions. Under a set of nominal viewing conditions, these parameters are set to 1,

and the overall effect is dictated solely by S_C and S_H, which reduce the perceived color difference as chroma increases.

Another metric used widely in the textile industry is the CMC color difference formula. This formula is similar in form to the ΔE^*_{94} equation and has parameters tailored for perceptibility vs. acceptability of color differences. Finally, an extension of the ΔE^*_{94} formula has been recently developed, known as the CIEDE2000 metric.[18] This metric accounts for interactions between the C^* and H^* terms and is expected to be adopted as an industry standard until further developments arise. The reader is referred to Chapter 1 for details.

The next question to consider is what error statistics to report. Common aggregate statistics cited in the literature are the mean, standard deviation, minimum, and maximum of the ΔEs for a set of test samples. Often, a cumulative statistic such as the 95th percentile of ΔE values (i.e., the value below which 95% of the ΔE values in the test data lie) is calculated. For a complete statistical description, histograms of ΔE can also be reported.

Having chosen an error metric, how does one determine that the characterization error is satisfactorily small? First, recall that characterization accuracy is limited by the inherent stability and uniformity of a given device. If the errors are close to this lower bound, we know that we cannot do much better for the given device. In the following sections, we will provide the reader with some idea of the characterization accuracy achievable by state-of-the-art techniques. It must be kept in mind, however, that "satisfactory accuracy" depends strongly on the application and the needs and expectations of a user. A graphic arts color proofing application will likely place stringent demands on color accuracy, while inexpensive consumer products will typically play in a market with wider color tolerances.

Another aspect that further confounds evaluation of color accuracy is that the end user ultimately views not test targets with color patches but images with complex color and spatial characteristics. Unfortunately, quantitative analysis of patches is not always a reliable indicator of perceived color quality in complex images. (The latter is a subject of active research.[19]) The reader is thus advised to exercise appropriate caution when interpreting individual results or those cited in the literature, and to always augment quantitative evaluation of color accuracy with a qualitative evaluation involving images and individuals that represent the intended market and application.

A special class of error metrics for input devices evaluates how accurately the information recorded by the input device can be transformed into the signals sensed by the human visual system for input stimuli with given spectral statistics. Such error metrics do not directly evaluate the accuracy of a characterization but rather the ability of the device to act as a visual colorimeter. Hence, these metrics are relevant for filter design optimization and can also suggest the most appropriate characterization technique for a given input device. The reader is referred to papers by Sharma et al.[20] and Quan et al.[21] for further details.

5.6 Scanners

All scanners employ one of two primary types of sensing technology. Drum scanners use photomultiplier tubes (PMTs), whereas the less expensive flatbed scanners employ charge-coupled devices (CCDs). Both of these technologies sense and convert light input into analog voltage. Drum scanners consist of a removable transparent cylinder on which a print, which is reflective, transparent, or a photographic negative, can be mounted. A light source illuminates the image in a single pass as the drum spins at a high speed. The light reflected off or transmitted through the print is passed through red, green, and blue filters then sent through the PMTs, which relay voltages proportional to the input light intensity. The PMT is extremely sensitive, thus providing drum scanners a large dynamic range. The drum scanners used in offset printing applications contain built-in computers that are capable of direct conversion of the RGB scan to CMYK output and are used to generate color separations at very high spatial resolution. A limitation of this scanning technology is that the original must be flexible so that it can physically be mounted on the drum.

All flatbed scanners utilize CCD technology, which is simpler, more stable, and less costly than PMT technology. These scanners have widely varying sensitivity and resolution and, at the highest end, approach the performance of drum scanners. Transparent or reflective prints are placed on a glass platen and evenly illuminated from above the glass for transparencies, and from beneath for reflective. As the light source moves across the image, individual lines of the image are sensed by a CCD array, which relays voltages that are proportional to the input light intensity. An integrating cavity is usually employed to focus light from the scanner illuminant onto the print. An undesirable outcome of this is that light reflected from a given spatial location on the print can be captured by the cavity and returned to the print at neighboring locations. Hence, the scanner measurement at a pixel depends not only on the reflectance at that pixel but also on the reflectances of neighboring pixels. A model and correction algorithm for this so-called *integrating cavity effect* is given by Knox.[22]

Following the sensing step, an analog-to-digital (A/D) converter is used to quantize the analog voltage signal to a digital signal represented by between 8 and 16 bits per each of R, G, B channels. These raw digital values are usually linear with respect to the luminance of the stimulus being scanned. Additional image acquisition software often allows the raw data to be processed through tone reproduction curves so that a power-law (or gamma) relationship exists between digital value and luminance. This operation is carried out before the A/D conversion. One reason for doing this is that quantization of nonlinear gamma-corrected signals is less visually disturbing than quantization of data that is linear in luminance. (This is discussed in more detail in the section on display characterization.) A second reason is to prepare the scanned data for direct display on a CRT, which exhibits approximately a square law (gamma = 2) relationship.

5.6.1 *Calibration*

Scanner calibration involves first establishing various settings internal to the scanner, or in the scanner driver. To calibrate the white point, a reflective white sample shipped with the scanner is scanned, and the gain factor on each of the R, G, B signals is adjusted so that R = G = B = 1 for this sample. As mentioned earlier, additional scanner software can offer selections for the digital precision of the RGB output and transformations between analog and digital representations (e.g., power-law functions). Once set, these parameters must not be altered during subsequent characterization or scanning operations.

In addition, it is usually desirable to linearize and gray-balance the scanner response. The result of this step is that an input ramp of gray stimuli in equal increments in luminance will result in equal increments in R = G = B scanner values. To achieve this, the scanner is exposed to a ramp of gray patches of known luminance values (e.g., as found at the bottom of the Q60 target); the scanner RGB values are extracted for each patch, and a TRC is constructed. A hypothetical example is given in Figure 5.23 to illustrate the process. The TRC is constructed so that a triplet of raw RGB values corresponding to a gray patch will map to the corresponding measured luminance value (within a scaling factor). The measurements generally provide only a subset of the data points in the TRC, the rest being determined with some

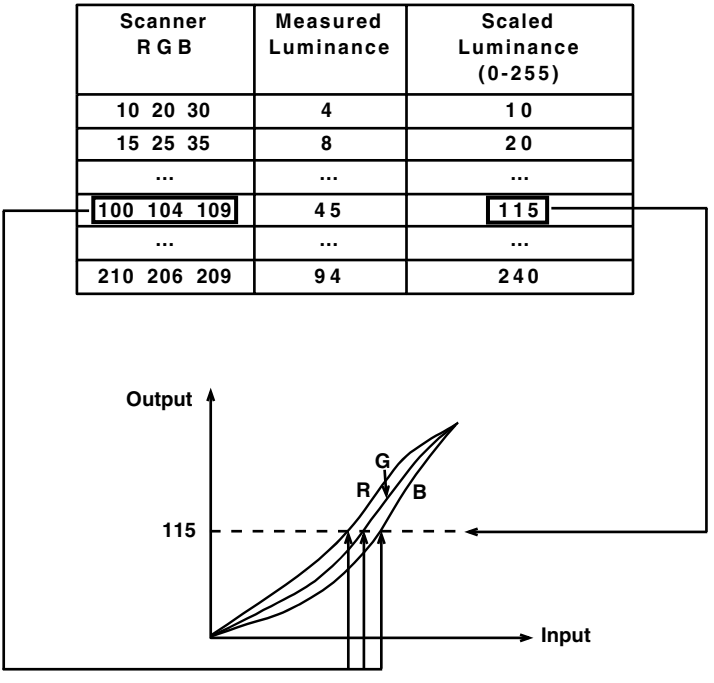


Figure 5.23 Illustration of gray-balance calibration for scanners.

form of data fitting or interpolation technique. Because the data are likely to contain some noise from the scanning and measuring process, it is preferable that the fitting technique incorporate some form of smoothing. Kang[23] reports that linear regression provides sufficiently accurate results for scanner gray balance, while nonlinear curve fitting offers only a modest improvement. In any event, polynomial and spline techniques are viable alternatives for scanners that exhibit significant nonlinearity.

5.6.2 Model-based characterization

Model-based scanner characterization attempts to establish the relationship between calibrated device-dependent data and colorimetric representations via explicit modeling of the device spectral sensitivities. Adopting the notation in previous sections, consider a training set of T spectral reflectance samples $\{\mathbf{s}_i\}$, which can be collected into a matrix $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_T]^t$. The spectral data is related to device data $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_T]^t$ and colorimetric data $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_T]^t$ by Equations 5.1 and 5.2, respectively. In matrix notation, we thus have

$$\mathbf{C} = \mathbf{S}\mathbf{A}_c \;;\; \mathbf{D} = \mathbf{S}\mathbf{A}_d \tag{5.38}$$

The column vectors of matrix $\mathbf{A}_c$ are a product of the color matching functions and the viewing illuminant $\mathbf{I}_v$, and similarly $\mathbf{A}_d$ is formed from a product of the scanner spectral sensitivities and the scanner illuminant $\mathbf{I}_s$. The classic model-based approach is to compute the linear 3×3 matrix transformation $\mathbf{M}$ that best fits the colorimetric data to device-dependent data in the least-squared error sense. The linear approximation is expressed as

$$\mathbf{C} \approx \mathbf{D} \cdot \mathbf{M} \tag{5.39}$$

and from Section 5.4.1, the optimal $\mathbf{M}$ is the least-squares solution,

$$\mathbf{M} = (\mathbf{D}^t\mathbf{D})^{-1}\,\mathbf{D}^t\mathbf{C} \tag{5.40}$$

Plugging Equation 5.38 into Equation 5.40, we have

$$\mathbf{M} = (\mathbf{A}_d^t\mathbf{S}^t\,\mathbf{S}\mathbf{A}_d)^{-1}\,\mathbf{A}_d^t\mathbf{S}^t\,\mathbf{S}\mathbf{A}_c \tag{5.41}$$

Equation 5.41 tells us that the scanner characterization function is determined by

1. Color matching functions
2. Viewing and scanning illuminants $\mathbf{I}_v$ and $\mathbf{I}_s$
3. Spectral autocorrelation matrix $\mathbf{S}^t\mathbf{S}$ of the training samples
4. Scanner spectral sensitivities

Note that Equation 5.40 can be directly used to estimate M from a set of training samples $\{\mathbf{d}_i, \mathbf{c}_i\}$ without explicit knowledge of the spectral sensitivities. However, for accurate results, this empirical procedure would have to be repeated for each different combination of input reflectances **S** and viewing illuminants $\mathbf{I}_v$. The model-based formulation, Equation 5.41, allows prediction of the scanner response for arbitrary input reflectances and illuminants given the scanner sensitivities $\mathbf{A}_d$ and illuminant $\mathbf{I}_s$. The optimal **M** can be computed using Equation 5.41 without having to make repeated measurements for every combination of input media and illuminants.

Each of the quantities of interest in Equation 5.41 will now be discussed. Because the color matching functions $\mathbf{A}_c$ are known functions, they are not included in the discussion.

Viewing illuminant. In general, it is difficult to ascertain *a priori* the illuminant under which a given stimulus will be viewed. A common *de facto* assumption for viewing reflective prints is the Daylight 5000 (D50) illuminant. However, if it is known that images are to be viewed under a certain type of lighting, e.g., cool-white fluorescence or an incandescent lamp, then the corresponding spectral radiance should be used.

Scanning illuminant. Scanners typically employ a fluorescent source, hence the spectral radiance function will contain sharp peaks as shown in Figure 5.4. The spectral radiance function $I_s(\lambda)$ can be obtained from the scanner manufacturer or can be estimated from the training data. However, the peaks found in fluorescent sources can lead to unreliable estimates unless these are explicitly modeled.[24] Hence, it is generally preferable that this quantity be directly measured.

Scanner spectral sensitivities. Deriving the scanner sensitivities is the most challenging aspect of model-based characterization. Some scanner manufacturers supply such data with their products. However, the information may not be accurate, as filter characteristics often change with time and vary from one scanner to another. Direct measurement of the scanner sensitivities may be achieved by recording the scanner response to narrowband reflectance data. However, this is a difficult and expensive process and therefore impractical in most applications. The most viable alternative is to estimate the sensitivities from a training set of samples of known spectral reflectance. Several approaches exist for this and are briefly described below, along with references for further reading.

The most straightforward technique is to use least-squares regression to obtain the device sensitivity matrix $\mathbf{A}_d$. The objective is to find $\mathbf{A}_d$ that minimizes $\|\boldsymbol{D} - \boldsymbol{S}\boldsymbol{A}_d\|^2$. From the linear regression formulation in Section 5.4.2, we have

$$\mathbf{A}_d = (\mathbf{S}^t\,\mathbf{S})^{-1}\,\mathbf{S}^t\mathbf{D} \tag{5.42}$$

The problem with this approach is that, although the spectral reflectance data is L-dimensional, with L being typically between 31 and 36, the true dimensionality of the spectra of samples found in nature is significantly less.

(Studies have shown that the samples in the Macbeth chart can be accurately represented with as few as three basis functions.[25]) Alternatively phrased, the system of Equations 5.42 contains only a small number of significant eigenvalues. This results in the spectral autocorrelation matrix $\mathbf{S}^t\mathbf{S}$ being ill conditioned, in turn yielding unstable, noise-sensitive estimates of the sensitivity functions $\mathbf{A}_d$. One approach to mitigate this problem is to use only the eigenvectors corresponding to the few most significant eigenvalues of $\mathbf{S}^t\mathbf{S}$ in the solution of Equation 5.42. This so-called "principal eigenvector" (PE) method results in a solution that is far less noise sensitive than that obtained from Equation 5.42. The reader is referred to Sharma[24] for more details.

One problem with PE is that it does not exploit *a priori* information about the nature of the spectral sensitivity functions. We know, for example, that the spectral sensitivities are positive-valued and usually single-lobed functions. In the case where $\mathbf{A}_d$ only contains the passive filter and detector responses (i.e., the illuminant is not included), we also know that the functions are smooth. There are a number of ways to use these constraints to generate estimates of $\mathbf{A}_d$ that are superior to those achieved by PE. One approach is to define the aforementioned constraints as a set of linear inequalities and formulate the least-squares minimization as a quadratic programming problem. The latter can be solved using standard packages such as Matlab. The reader is referred to Finlayson et al.[26] for more details. Another approach is to use a set theoretical formulation to express the constraints as convex sets and to use an iterative technique known as *projection onto convex sets (POCS)* to generate the sensitivity functions.[24] One potential problem with the POCS technique is that the solution is not unique and is often sensitive to the initial estimate used to seed the iterative process. Despite this caveat, this technique has been shown to produce very good results.[24,27]

Input spectral data. As alluded to in Section 5.2, the spectral reflectance data **S** should be measured from media that are representative of the stimuli to be scanned. If a single scanner characterization is to be derived for all possible input media, it is advisable to measure the data from a wide range of media, e.g., photography, offset, laser, inkjet, etc. An interesting case occurs if **S** is constructed by drawing samples at random from the interval [–1, 1] with equal likelihood. With this "maximum ignorance" assumption, the spectral data are uncorrelated; therefore, the autocorrelation $\mathbf{S}^t\mathbf{S}$ is an identity matrix, and Equation 5.41 reduces to

$$\mathbf{M} = (\mathbf{A}_d{}^t\mathbf{A}_d)^{-1}\,\mathbf{A}_d{}^t\mathbf{A}_c \tag{5.43}$$

Note that the characterization transform now no longer depends on measured data. Observe, too, that Equation 5.43 is also the least-squares solution to the linear transformation that relates the color matching functions $\mathbf{A}_c$ to the device sensitivities $\mathbf{A}_d$;

$$\mathbf{A}_c \approx \mathbf{A}_d\mathbf{M} \tag{5.44}$$

Comparing Equations 5.39 and 5.44, we see that the optimal linear transform that maps the color matching functions to the scanner sensitivities is the same as the transform that optimally maps scanner RGB to XYZ under the maximum ignorance assumption. As a corollary, if the scanner is perfectly colorimetric, then Equations 5.39 and 5.44 become equalities, and the matrix that relates the color matching functions to scanner sensitivities is precisely the matrix that maps scanner RGB to XYZ for all media and illuminants.

One problem with the maximum ignorance assumption is that it includes negative values, which can never occur with physical spectra. Finlayson et al.[28] show that a positivity constraint on the preceding formulation results in the correlation $\mathbf{S}^t\mathbf{S}$ being a constant (but not identity) matrix, which results in a more accurate estimate of $\mathbf{M}$.

Another class of model-based techniques, somewhat distinct from the preceding framework, derives scanner characterization for a specific medium by first characterizing the medium itself and using models for both the medium and scanner to generate the characterization. The additional step of modeling the medium imposes physically based constraints on the possible spectra S and can lend further insight into the interaction between the medium and the scanner. Furthermore, *a priori* modeling of the input medium may simplify the *in situ* color measurement process. Berns and Shyu[29] postulate that scanner filters are designed to align closely with the peaks of the spectral absorptivity functions of typical photographic dyes. The relationship between scanner RGB and C, M, Y dye concentrations is thus modeled by simple polynomial functions. The Beer–Bouguer and Kubelka–Munk theories (discussed in Section 5.10.2) are then used to relate dye concentrations to reflectance spectra for photographic media. Sharma[30] models the color formation process on photographic media using the Beer–Bouguer model. From this model, and using a small number of measurements on the actual sample being scanned, the set $\mathbf{S}_{medium}$ of all reflectance spectra reproducible by the given medium is estimated. For a given scanner RGB triplet, the set $\mathbf{S}_{scanner}$ of all reflectance spectra that can generate this triplet is derived with knowledge of the scanner spectral sensitivities, $\mathbf{A}_d$. The actual input reflectance spectrum lies in the intersection $\mathbf{S}_{medium} \cap \mathbf{S}_{scanner}$ and is derived using POCS. Note that both these approaches generate spectral characterizations, i.e., mappings from scanner RGB to spectral reflectance. From this, colorimetric characterizations can readily be generated for arbitrary viewing illuminants.

5.6.3 *Empirical characterization*

Empirical approaches derive the characterization function by correlating measured CIE data from a target such as the Q60 to scanned RGB data from the target. Most of the data-fitting techniques described in Section 5.4 can be used (with the exception of lattice-based approaches, as scanner

characterization data cannot be designed to lie on a regular grid). Kang[23] describes the use of polynomial regression to fit gray-balanced RGB data to CIEXYZ measurements. He compares 3×3, 3×6, 3×9, 3×11, and 3×14 polynomial matrices derived using least-squares regression as described in Section 5.4.3. Several targets, including the MacBeth ColorChecker and Kodak Q60, are used. The paper concludes that a 3×6 polynomial offers acceptable accuracy and that increasing the order of the polynomial may improve the fit to training data but may worsen the performance on independent test data. This is because, as noted in Section 5.4, higher-order approximations begin to track the noise in the data. The paper also explores media dependence and concludes that the optimal 3×3 matrix does not vary considerably across media, whereas the optimal polynomial transform is indeed media dependent and will generally offer greater accuracy for any given medium.

Kang and Anderson[31] describe the use of neural networks for scanner characterization. They use a 3–4–3 network, trained by cascaded feed-forward correlation. A cumulative Gaussian function is used for the nonlinearity at each unit in the network (see Section 5.4.7). In comparison with polynomial regression, the neural network reports superior fits to training data but inferior performance for independent test data. Furthermore, the neural network is reported as being fairly sensitive to the choice of training data. Hence, while neural networks offer powerful capabilities for data fitting, much care must be exercised in their design and optimization to suit the nature of the particular device characteristics.

5.7 Digital still cameras

Digital still cameras (DSCs) are becoming a common source for digital imagery. Their characterization is complicated by two factors.

1. The conditions under which images are captured are often uncontrolled and can vary widely.
2. To compensate for this, DSC manufacturers build automatic image-processing algorithms into the devices to control and correct for flare, exposure, color balance, etc.

DSC characterization is probably unnecessary in most consumer applications and is called for only in specialized cases that require controlled, high-quality color capture. In such cases, it is imperative that the automatic processing be disabled or known to the extent that the raw DSC signals can be recovered from the processed data.

A few precautions are in order for proper digital capture of calibration and characterization targets. First, it must be ensured that the illumination on the target is uniform. A viewing/illuminating geometry of 0/45 is recommended so as to be consistent with the geometry of measurement devices and typical visual viewing of hardcopy prints. Next, the lenses in most digital

cameras do not transmit light uniformly across the lens area, so, for a fixed input radiance, pixels near the center report higher signal levels than those in the periphery. The ideal solution to this problem is to expose the camera to a constant color (e.g., gray) target and digitally compensate for any spatial uniformity in the camera response. (Such compensation may be built into some camera models.) The effect can also be somewhat reduced by choosing the distance between camera and target so that the target does not occupy the full camera frame. Finally, it is recommended that any nonvisible radiation to which the DSC is sensitive be blocked so that output RGB values are not affected. Many DSCs respond to IR radiation, hence IR blocking filters should be used.

Figure 5.24 shows the color calibration and characterization path for a DSC. Much of the theoretical framework for image capture is common between DSCs and scanners; hence, we will frequently refer to the formulation developed in Section 5.6 for scanners while focusing here on DSC-specific issues. For additional procedural details on DSC characterization, the reader is referred to the ISO 17321 standard.[32]

5.7.1 Calibration

It must be ensured that camera settings such as aperture size and exposure time are in a known fixed state, and that all automatic color processing is disabled. The main task in DSC calibration is to determine the relationship between input scene radiance and camera response, typically for a range of gray input stimuli. Determination of this function, known as the opto-electronic conversion function (OECF), is conceptually similar to the gray-balancing operation for a scanner (see Section 5.6.1). A target comprising gray patches of known spectral reflectance measurements is illuminated with a known reference illuminant. From the reflectance and illuminant data, the luminance Y of each patch is calculated (see Equation 5.9). An image of the target is captured with the DSC. The correspondence between input luminance Y and output RGB is used to generate an inverse OECF function as described in Section 5.6.1 for scanners. This is a TRC that maps raw device

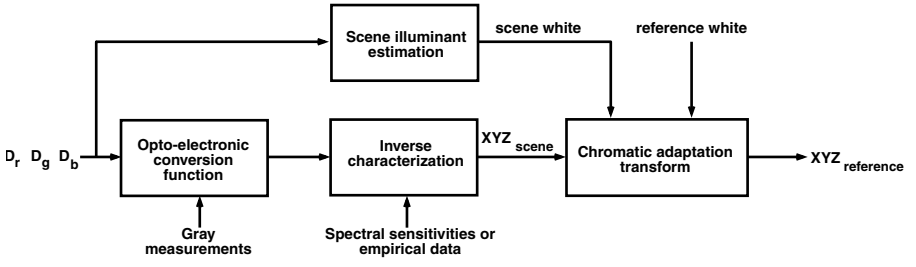


Figure 5.24 Block diagram of digital camera calibration, characterization, and chromatic adaptation transforms.

RGB to R′G′B′ such that R′ = G′ = B′ = Y for the neutral patches. The raw captured image is then always processed through this TRC to obtain a linearized and gray-balanced image prior to subsequent processing. Further details on specifications and experimental procedures for OECF determination are given in the ISO 14524 standard.[33]

5.7.2 *Model-based characterization*

The goal is to obtain the optimal 3×3 matrix **M** that relates the DSC RGB data to a colorimetric (e.g., XYZ) representation. As with scanners, derivation of **M** is given by Equation 5.41 and requires knowledge of the color matching functions, correlation statistics of scene data, and device spectral sensitivities. Color matching functions are known and require no further discussion. Scene correlation statistics should be used where possible. However, given the diversity of scene content likely to be encountered by a DSC, the maximum ignorance assumption is often invoked, and scene statistics are eliminated from the formulation. Derivation of **M** thus reduces to Equation 5.43 and requires only estimation of the DSC spectral sensitivities.

The techniques described in Section 5.6.2 for estimating device sensitivities indirectly from the characterization data can be applied for DSCs. One can also adopt a more direct approach of recording the device's response to incident monochromatic light at different wavelengths. The latter can be generated by illuminating a diffuse reflecting surface with light filtered through a monochromator. From Equation 5.1, the camera response to monochromatic light at wavelength λ is given by

$$D_i(\lambda) = I_m(\lambda)R_d(\lambda)q_i(\lambda) = S(\lambda)q_i(\lambda) \tag{5.45}$$

where $i = R, G, B$
$I_m(\lambda)$ = the monochromator illumination
$R_d(\lambda)$ = the reflectance of the diffuse surface
$S(\lambda) = I_m(\lambda)R_d(\lambda)$ is the radiance incident to the DSC

For simplicity, the detector sensitivity $u(\lambda)$ in Equation 5.1 is folded into the term $q_i(\lambda)$ in Equation 5.45, and the noise term is assumed to be negligible. The radiance $S(\lambda)$ is measured independently with a spectroradiometer. The spectral sensitivities $q_i(\lambda)$ are then obtained by dividing the camera response $D_i(\lambda)$ by the input radiance $S(\lambda)$. In the case where the DSC response is tied to a specific reference illuminant $I_{ref}(\lambda)$, the products $q_i(\lambda)I_{ref}(\lambda)$ can be stored. More details are found in ISO 17321.[32]

The reader is reminded that, due to practical considerations, DSC sensitivities are not linearly related to color matching functions, and that the 3×3 matrix being derived is only an approximation. However, this approximation is sufficient for many applications. The accuracy of **M** for critical colors can be further improved by imposing constraints on preservation of white and neutral colors.[34]

5.7.3 Empirical characterization

As with scanners, empirical DSC characterization is accomplished by directly relating measured colorimetric data from a target and corresponding DSC RGB data obtained from a photographed image of the target. This approach is recommended in the case where the DSC spectral sensitivities are unknown, or when the target and illumination conditions used for characterization are expected to closely match those encountered during actual image capture.

Hubel et al.[35] compare several techniques for computing the optimal 3×3 matrix **M**. One of these is a model-based approach that uses a white point preserving maximum ignorance assumption, while the remaining techniques are empirical, using linear regression on training samples. They report an extensive set of results for different illumination conditions. Average ΔE_{CMC} values range from approximately 2.5 to 6.5, depending on the technique and illumination used. The model-based technique was often outperformed by an empirical technique for a given medium and illuminant. However, the model-based strategy, being oblivious to scene statistics, was generally robust across different illumination conditions.

An empirically derived characterization need not be restricted to a linear transformation. Hong et al.[36] explore a polynomial technique to characterize a low-performance Canon PowerShot Pro70 camera for photographic input. A second-order polynomial was employed with 11 terms given by [D_r, D_g, D_b, D_rD_g, D_rD_b, D_gD_b, D_r^2, D_g^2, D_b^2, $D_rD_gD_b$, 1]. The average characterization error for 264 training samples from an IT8.7/2 target was $\Delta E_{CMC(1:1)} = 2.2$. A similar technique[37] was used to characterize a high-performance Agfa digital StudioCam resulting in an average $\Delta E_{CMC(1:1)} = 1.07$. Note that these errors are significantly lower than those reported by Hubel et al. This is not surprising, because polynomials can be expected to outperform linear approximations under a given set of controlled characterization conditions. The other findings from these two studies are as follows:

- Correction for the OECF significantly improves overall characterization accuracy.
- For polynomial fitting, 40 to 60 training samples seem adequate; beyond this, there is little to be gained in characterization accuracy.
- The polynomial correction is highly dependent on the medium/colorant combination.
- For a single medium/colorant combination, increasing the order of the polynomial up to 11 improves the characterization accuracy, with some terms (notably $D_rD_gD_b$ and the constant term) being more important than others. With the high-performance camera, a 3×11 polynomial results in an average error of approximately 1 $\Delta E_{CMC(1:1)}$. The low-performance camera results in $\Delta E_{CMC(1:1)} = 2.2$.
- For cross-media reproduction, increasing the order of the polynomials is not of significant benefit. Typical accuracy with a 3×11

correction lies between 2 and 4 $\Delta E_{CMC(1:1)}$ when characterization and test media are not the same.

5.7.4 White-point estimation and chromatic adaptation transform

The characterization step described in Sections 5.7.2 and 5.7.3 yields a transformation between DSC data and colorimetric values corresponding to the input viewing conditions. One must be able to convert this colorimetric data to a standard color space (e.g., sRGB), which is based on a different set of reference viewing conditions. This calls for a color appearance model to account for the differences between input and reference viewing conditions. The most important parameters pertaining to the viewing conditions are the input scene and reference white points. The appearance model can thus be reduced to a chromatic adaptation transform (CAT) between the two white points.

In general, the scene white is unknown and must be indirectly estimated from the image data. A recent technique, known as color by correlation, has shown promise as a simple and reliable method of estimating white point. The idea is to acquire *a priori* sets of DSC training data corresponding to different known illuminants. Data from a given image are then compared with each training set, and the illuminant is chosen that maximizes the correlation between the image and training data. If the DSC spectral sensitivities are known, the training samples can be acquired via simulation; otherwise, they must be gathered by photographing samples under different illuminants. See Chapter 5 of Reference 7 for details of this approach.

There has been considerable research in finding the optimal color space for the CAT. An excellent survey is given in Chapter 5 of Reference 7. Ideally, the CAT should mimic visual adaptation mechanisms, suggesting that it should be performed in an LMS cone fundamental space. Finlayson et al.[38] use the added argument that orthogonal visual channels maximize efficiency to orthogonalize the LMS space, forming their so-called *sharp* color space. (The term "sharp" comes from the fact that the associated color matching functions are relatively narrowband.) Psychophysical validation has shown that the sharp space is among the best spaces for performing the CAT. A physically realizable variant of this space is being proposed as an ISO standard for DSC characterization.[32] This ISO-RGB space is a linear transformation of XYZ, and is given by

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4339 & 0.3762 & 0.1899 \\ 0.2126 & 0.7152 & 0.0721 \\ 0.0177 & 0.1095 & 0.8728 \end{bmatrix} \begin{bmatrix} R \\ G \\ G \end{bmatrix}; \quad \begin{bmatrix} R \\ G \\ B \end{bmatrix} \begin{bmatrix} 3.0799 & -1.5369 & -0.5432 \\ -0.9209 & 1.8756 & 0.0454 \\ 0.0531 & -0.2041 & 1.1510 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}; \tag{5.46}$$

The procedure for applying the CAT in ISO-RGB space given the input and reference white points is summarized as follows:

1. Use the calibration and characterization transforms to convert DSC device data to XYZ_{in} corresponding to input viewing conditions.
2. Convert the input and reference white points from XYZ to ISO-RGB using Equation 5.46.
3. Convert XYZ_{in} to ISO-RGB_{in} using Equation 5.46.
4. Perform von Kries chromatic adaptation by multiplying ISO-RGB_{in} by the diagonal matrix,

$$\begin{bmatrix} \frac{R_{ref}^{white}}{R_{in}^{white}} & 0 & 0 \\ 0 & \frac{G_{ref}^{white}}{G_{in}^{white}} & 0 \\ 0 & 0 & \frac{B_{ref}^{white}}{B_{in}^{white}} \end{bmatrix} \tag{5.47}$$

   where C_{ref}^{white}, C_{in}^{white}, $(C = R, G, B)$ are the white points under reference and input viewing conditions, respectively. (The reader is referred to an earlier chapter for details on von Kries adaptation.) This step generates ISO-RGB data under reference viewing conditions, denoted ISO-RGB_{ref}.
5. Convert ISO-RGB_{ref} to XYZ_{ref} using Equation 5.46. This provides a colorimetric representation under reference viewing conditions and can be transformed to other standard color spaces.

Note that the matrices in the last three steps can be concatenated into a single 3×3 matrix for efficient processing.

5.8 CRT displays

The cathode-ray tube (CRT) is the most common type of display used in computers and television. Color is produced on a CRT display by applying modulated voltages to three electron guns, which in turn strike red, green, and blue phosphors with electrons. The excited phosphors emit an additive mixture of red, green, and blue lights. The assumptions mentioned in Section 5.2.4 on channel independence and chromaticity constancy, in addition to the usual assumptions on spatial uniformity and temporal stability, result in a fairly simple process for CRT calibration and characterization.

5.8.1 Calibration

Cathode-ray tube (CRT) calibration involves setting brightness and contrast controls on the display to a fixed nominal value. In addition, the relationship between the R, G, B input digital values driving the three gun voltages and

the resulting displayed luminance must be established and corrected. This relationship is usually modeled based on the power-law relationship between the driving voltage and the beam-current for a vacuum tube, and is given by[39]

$$\frac{Y_R}{Y_{R_{max}}} = \begin{cases} f + K_R\left(\dfrac{D_R - D_{offset}}{D_{max} - D_{offset}}\right)^{\gamma_R} & \text{if } D_R > D_{offset} \\ f & \text{if } D_R \le D_{offset} \end{cases} \tag{5.48}$$

where D_R = input digital value to the red gun
Y_R = resulting luminance from the red channel
$Y_{R_{max}}$ = luminance of the red channel at full intensity
D_{offset} = largest digital count for which there is no detectable luminance from the screen
D_{max} = maximum digital count (e.g., in an 8-bit system, $D_{max} = 255$)
f = flare that arises mostly from ambient illumination
K_R = a gain factor
γ_R = nonlinear power law factor

Analogous expressions hold for the green and blue terms. Generally, the calibration is done with all room lights turned off; hence, the flare term is assumed to be negligible. In addition, with proper brightness and contrast settings, the following simplifying assumptions are often made: $K_R = K_G = K_B = 1$, $D_{offset} = 0$. This reduces Equation 5.48 to

$$\frac{Y_R}{Y_{R_{\max}}} = \left(\frac{D_R}{D_{\max}}\right)^{\gamma_R} \tag{5.49}$$

with analogous expressions for Y_G and Y_B. The parameters for the calibration model are obtained by making measurements of a series of stepwedges from each primary color to black using a spectroradiometer or colorimeter and fitting these measurements to the model given by Equations 5.48 or 5.49 using regression. If Equation 5.49 is adopted, a simple approach is to take logarithms of both sides of this equation to produce a linear relationship between $\log(Y_R)$ and $\log(D_R/D_{max})$. This can then be solved for γ_R via the linear regression technique described in Section 5.4.1. Berns et al.[39] provide detailed descriptions of other regression techniques. Values of γ_R, γ_G, γ_B for typical CRTs lie between 1.8 and 2.4.

Once the model is derived, a correction function that inverts the model is applied to each of the digital R, G, B inputs. If Equation 5.49 is assumed, the correction is given by

$$D_R = D_{\max}\left(\frac{D'_R}{D_{\max}}\right)^{1/\gamma_R} \tag{5.50}$$

with similar expressions for G and B. Here D'_R, D'_G, D'_B are linear in luminance, and D_R, D_G, D_B are the raw signals that drive the gun voltages. The calibration function, Equation 5.50, is often referred to as gamma correction and is usually implemented as a set of three one-dimensional lookup tables that are loaded directly into the video path. Plots of Equations 5.49 and 5.50 for $\gamma = 1.8$ are shown in Figure 5.25.

It is worth noting that digital quantization of the gamma-corrected signal D_R, D_G, D_B in Equation 5.50 results in smaller quantization intervals at lower luminance values where the eye is more sensitive to errors, and larger intervals at high luminances where the eye is less sensitive. The idea of applying nonlinear preprocessing functions to reduce the visual perceptibility of quantization errors (often known as *companding*) is widely employed in many digital signal processing applications. In our case, gamma correction applied prior to conversion to the digital domain not only calibrates the CRT, it also fortuitously reduces perceived quantization error in color images intended for CRT display.

The CRT with the gamma correction Equation 5.50 incorporated in the video path exhibits a tone reproduction characteristic that is linear in luminance. That is,

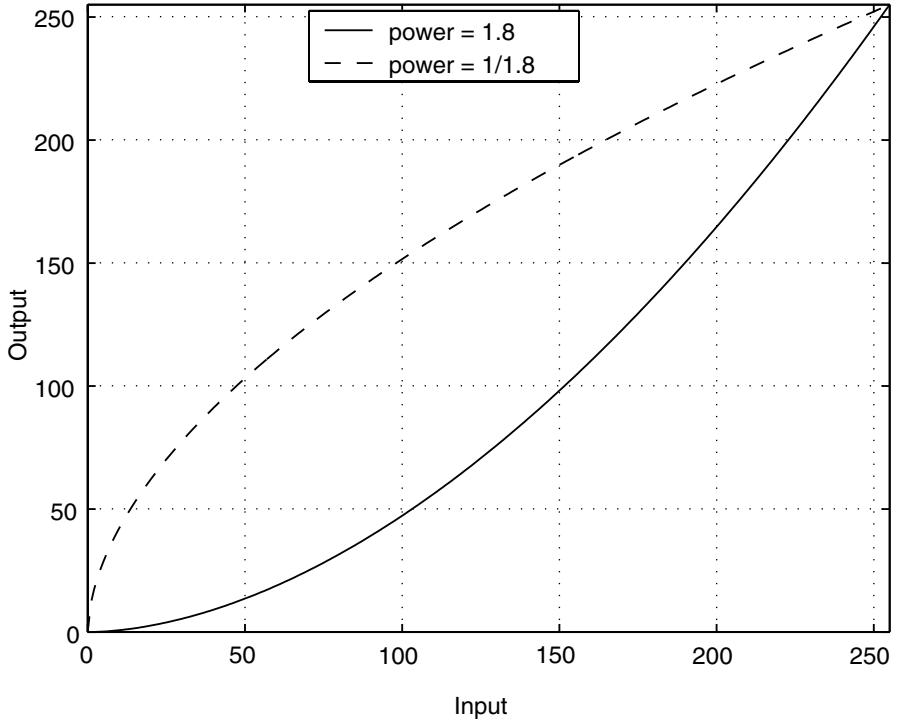


Figure 5.25 Gamma function for $\gamma = 1.8$.

$$\frac{Y_R}{Y_{R_{\text{Max}}}} = \left(\frac{D'_R}{D_{\text{max}}}\right) \tag{5.51}$$

with similar expressions for G and B. Some CRT calibration packages allow the user to specify an overall system gamma, γ_{system}, so that Equation 5.51 becomes

$$\frac{Y_R}{Y_{R_{max}}} = \left(\frac{D'_R}{D_{max}}\right)^{\gamma_{system}} \tag{5.52}$$

This provides some control on the effective tone reproduction characteristic of the CRT. To achieve this overall system response, the gamma correction function Equation 5.50 is modified as

$$D_R = D_{max}\left(\frac{D'_R}{D_{max}}\right)^{\left(\frac{\gamma_{system}}{\gamma_R}\right)} \tag{5.53}$$

5.8.2 Characterization

We assume henceforth that the aforementioned calibration has been derived so that Equation 5.51 holds. Recall that, with the assumptions on channel independence and chromaticity constancy, Equation 5.8 describes the relationship between input device RGB values and output spectral radiance. Spectral radiance is then converted to tristimulus XYZ values according to Equation 5.9. Substituting the expression for $S_{RGB}(\lambda)$ in Equation 5.8 into Equation 5.9., the relationship between the inputs D'_R, D'_G, D'_B to a linearized CRT and resulting tristimulus values is given by

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_R & X_G & X_B \\ Y_R & Y_G & Y_B \\ Z_R & Z_G & Z_B \end{bmatrix} \begin{bmatrix} D'_R \\ D'_G \\ D'_B \end{bmatrix} \tag{5.54}$$

where X_R, Y_R, Z_R = tristimulus values of the red channel at its maximum intensity, and likewise for green and blue

In matrix-vector notation, Equation 5.54 becomes

$$\mathbf{c} = \mathbf{A}_{CRT}\,\mathbf{d'}; \qquad \mathbf{d'} = \mathbf{A}_{CRT}^{-1}\,\mathbf{c} \tag{5.55}$$

The columns of $\mathbf{A}_{CRT}$ are the tristimulus coordinates of R, G, B at maximum intensity and can be obtained by direct tristimulus measurement. A more robust approach would be to include additional tristimulus measurements of other color mixtures and to solve for $\mathbf{A}_{CRT}$ using least-squares regression as described in Section 5.4.1. Note that $\mathbf{A}_{CRT}$ assumes flare-free viewing conditions. If flare is present, this can be captured in the **d'** vector by using calibration function Equation 5.48 with an appropriate value for *f*.

The quality of the characterization can be evaluated by converting a test set of color patches specified in XYZ to display RGB through the inverse characterization mapping (i.e., the second part of Equation 5.55) and measuring the displayed colors (see Figure 5.10). The original and measured values are then converted to CIELAB coordinates, and the error is derived using a suitable metric such as ΔE^*_{ab} or ΔE^*_{94}. Berns et al.[39] report excellent results using this simple model, with average ΔE^*_{ab} less than 1. Factors that can contribute to additional errors include internal flare within the CRT, cross-channel interactions not accounted for in the aforementioned model, and spatial nonuniformity across the display.

Most CRTs exhibit fairly similar color characteristics, because the power-law relationship, Equation 5.48, is a fundamental characteristic of vacuum tube technology; furthermore, CRT manufacturers use very similar, if not identical, phosphor sets. For this reason, and because CRTs are such a prevalent medium for the display and manipulation of color, there have been several efforts to standardize on CRT RGB color spaces. The most notable recent example is the sRGB standard (available at www.srgb.com). If one's CRT has not been characterized, one of the standard models can be adopted as a reasonable approximation. Minimally, these RGB spaces are defined by a gamma (assumed to be equal for all channels) and matrix $\mathbf{A}_{CRT}$. Sometimes, instead of directly specifying $\mathbf{A}_{CRT}$, the *x*-*y* chromaticity coordinates of the red, green, and blue primaries are provided along with the XYZ values of the white point. $\mathbf{A}_{CRT}$ is easily derived from these quantities (see Appendix 5.B).

5.8.3 *Visual techniques*

Because CRTs can be accurately characterized with simple models, a class of techniques has emerged that obviates the need for color measurements and relies upon visual judgments to directly estimate model parameters such as gamma and offset.[40–42] The basic idea is to display a series of targets on the screen and provide the user with some control to adjust certain colors until they match given reference stimuli. Based on the settings selected by the user, an algorithm computes the model parameters. An example is shown in Figure 5.26 for visually determining γ in Equation 5.49. The bottom half of the target is a fine checkerboard pattern of alternating black and white dots. The top half is a series of patches at different gray levels. The user is asked to select the gray patch whose luminance matches the average luminance of the checkerboard. The assumption is that the average checkerboard luminance $Y_{checkerboard}$ is approximately halfway between the luminances of

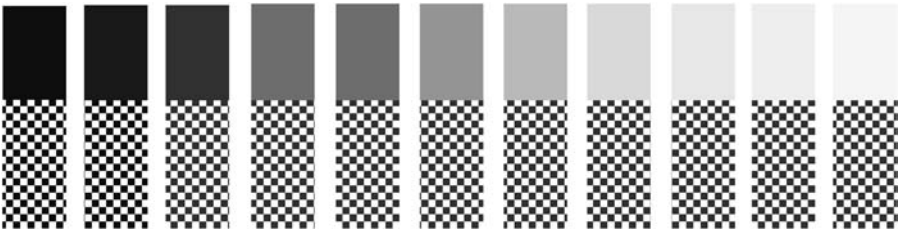


Figure 5.26 Target for visual determination of γ for displays.

black and white. Reasonable *a priori* assumptions can be made for the latter (e.g., $Y_{black} = 0$ and $Y_{white} = 100$, respectively), and hence for the checkerboard (e.g., $Y_{checkerboard} = 50$). A user who selects the gray patch to match the checkerboard is effectively selecting the digital count D_{match} corresponding to luminance $Y_{checkerboard}$. This provides enough information to calculate γ by rearranging Equation 5.49 as follows:

$$\gamma = \frac{\log(Y_{checkerboard}/Y_{white})}{\log(D_{match}/D_{max})} \tag{5.56}$$

In this example, the same γ value is assumed for the R, G, and B channels. The technique is easily extended to estimate γ for each individual channel by displaying checkerboard patterns that alternate between black and each respective primary. A demonstration of visual CRT calibration can be found in the recent article by Balasubramanian et al.[43] Visual determination of the color of the primaries and white point (i.e., $\mathbf{A}_{CRT}$) requires more sophisticated techniques[44] and is an active area of research.

5.9 Liquid crystal displays

Liquid crystal displays are becoming an increasingly popular medium for color display. Their compactness and low power consumption, combined with steadily increasing spatial resolution and dynamic range, have made these devices increasingly prevalent in both consumer and professional markets. Consequently, color management for LCDs has received greater attention in recent years.

The type of LCD most commonly used for computer display is the backlit active-matrix LCD (AMLCD) employing twisted nematic technology. In this technology, each pixel comprises a pair of linear polarizers and a liquid crystal substrate sandwiched between them. The polarizations are oriented orthogonally to each other. Light from a source behind the display surface passes through the first polarizer and is then reoriented by the liquid crystal substrate before it is passed through the second polarizer. The light then passes through one of red, green, or blue filters, arranged in a spatial mosaic. The extent of optical reorientation by the liquid crystal, and thus the intensity

of light finally emanated, is determined by an electric field applied to the liquid crystal substrate. This field is determined by an applied voltage, which in turn is driven by the digital input to the device.

From the viewpoint of color characterization, twisted nematic technology can pose several shortcomings: the strong dependence of perceived color on viewing angle, poor gray balance for R = G = B input, and lack of chromaticity constancy. Recent developments such as in-plane switching technology[45] overcome these problems to some extent.

5.9.1 *Calibration*

A major difference between CRT and LCD characteristics is the nonlinear function that relates input digital values to output luminance, shown in Figure 5.27. Unlike vacuum tubes that exhibit a power-law relationship, LCD technology results in a native electro-optic response that is often better modeled as a sigmoidal S-shaped function.[46] However, many LCD manufacturers build correction tables into the video card that result in the LCD response mimicking that of a CRT (i.e., a power-law response with $\gamma = 1.8$ or 2.2). Hence, it is recommended that some initial analysis be performed before a particular function is chosen and that, if possible, built-in corrections be deactivated so as to reliably calibrate the raw display response. As with CRTs, the calibration function is derived by making color measurements of a series of stepwedges in each of R, G, B. If a model-based approach is adopted, the model parameters are fitted to the measurements via regression.

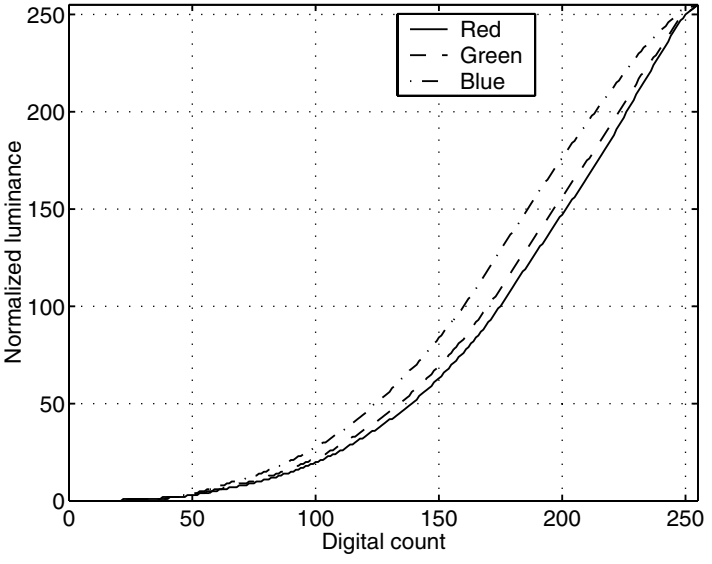


Figure 5.27 Typical opto-electronic conversion function for liquid crystal displays.

Alternatively, if the LCD response does not appear to subscribe to a simple parametric model, an empirical approach may be adopted wherein the measured data are directly interpolated or fitted using, for example, piecewise linear, polynomial, or spline functions.

As mentioned earlier, some LCDs do not adhere to the chromaticity constancy assumption. This is largely due to the non-smooth spectral characteristics of the backlight and its interaction with the color filters.[45] Kwak et al.[47] compensate for the lack of chromaticity constancy by introducing cross terms in the nonlinear calibration functions to capture interactions among R, G, and B. They claim a significant improvement in overall accuracy as a result of this extension.

5.9.2 *Characterization*

Most of the assumptions made with CRTs (i.e., uniformity, stability, pixel independence, and channel independence) hold to a reasonable degree with AMLCDs as well. Hence, the characterization function can be modeled with a 3×3 matrix as in Equation 5.54, and the procedure described in Section 5.8 for deriving CRT characterization can be used for AMLCDs in the same manner. As mentioned earlier, an important caution for AMLCDs is that the radiance of the emanated light can be a strong function of the viewing angle. The only practical recommendation to mitigate this problem is that the measurements should be taken of light emanating perpendicular to the plane of the screen. The same geometry should be used for viewing images. For further details on LCD characterization, the reader is referred to the works by Marcu,[45] Sharma,[46] and Kwak.[47]

5.10 *Printers*

Printer characterization continues to be a challenging problem due to the complex nonlinear color characteristics of these devices. Space considerations do not permit a description of the physics of the numerous digital printing technologies. Instead, we will offer general techniques that apply to broad categories of devices (e.g., halftone vs. continuous tone; CMY vs. CMYK, etc.).

Recall the basic calibration and characterization workflow in Figure 5.9. The techniques for target generation and calibration and characterization vary widely, offering a range of trade-offs between cost and accuracy. A selection of common techniques will be presented in this section.

5.10.1 *Calibration*

Two common approaches are channel-independent and gray-balanced calibration.

5.10.1.1 Channel-independent calibration

In this type of calibration, each channel i (i = cyan, magenta, yellow, etc.) is independently linearized to a defined metric M_i. An example of such a metric is the ΔE^*_{ab} color difference between the ith channel and medium white, defined as

$$M_i(d) = \|\mathbf{c}_{medium} - \mathbf{c}_i(d)\|_2, \quad i = C,M,Y, 0 \le d \le d_{max} \tag{5.57}$$

where d = input digital level
$\mathbf{c}_{medium}$ = CIELAB measurement of the medium
$\mathbf{c}_i(d)$ = CIELAB measurement of the ith colorant generated at digital level d

Note that, by definition, $M_i(0) \equiv 0$. Linearizing with respect to this metric will result in an approximately visually linear printer response along each of its primary channels.

The calibration is accomplished with the following steps:

- Generate stepwedges of pure C, M, Y patches at a few selected digital levels d_j. The number and spacing of levels required depend on the characteristics of the printer. As a general guideline, between 15 and 20 patches per channel is sufficient for most printers, and a finer sampling is recommended in the region of small d values to accurately capture the printer response at the highlights. Also ensure that the solid patch (i.e., $d = d_{max}$) is included.
- Make CIELAB measurements of the stepwedges and of the bare medium. Media relative colorimetry is recommended for the CIELAB calculations.
- Evaluate $M_i(d_j)$ at the measured digital levels d_j using Equation 5.57.
- Scale the data by a multiplicative factor so that $M_i(d_{max}) = d_{max}$. This is accomplished by multiplying the function $M_i(d)$ by the constant $[d_{max}/M_i(d_{max})]$.
- Invert the scaled functions $M_i(d)$ to obtain M_i^{-1} by interchanging the dependent and independent variables. Use some form of fitting or interpolation to evaluate M_i^{-1} for the entire domain $[0, d_{max}]$. If the printer response is smooth, linear interpolation suffices; otherwise, more sophisticated fitting techniques such as polynomials or splines are called for (see Section 5.4.). The result is the calibration function, which can be implemented as a set of one-dimensional TRCs for efficient processing of images.
- Test the calibration by running a stepwedge of uniformly spaced digital values of a single colorant through the TRC, printing and measuring the resulting patches, and computing M_i. A linear relationship should be achieved between the digital input to the TRC and the resulting M_i. Repeat this step for each colorant.

An example of the response $M_i(d)$ for a Xerox DocuColor 12 xerographic printer is shown in Figure 5.28a for 16 digital levels. The scaled $M_i(d)$ are shown in Figure 5.28b. The inverse function is shown in Figure 5.29 and is the final calibration TRC for the DC12. Note that the calibration is essentially a reflection of the printer response $M_i(d)$ about the 45° line. To test the

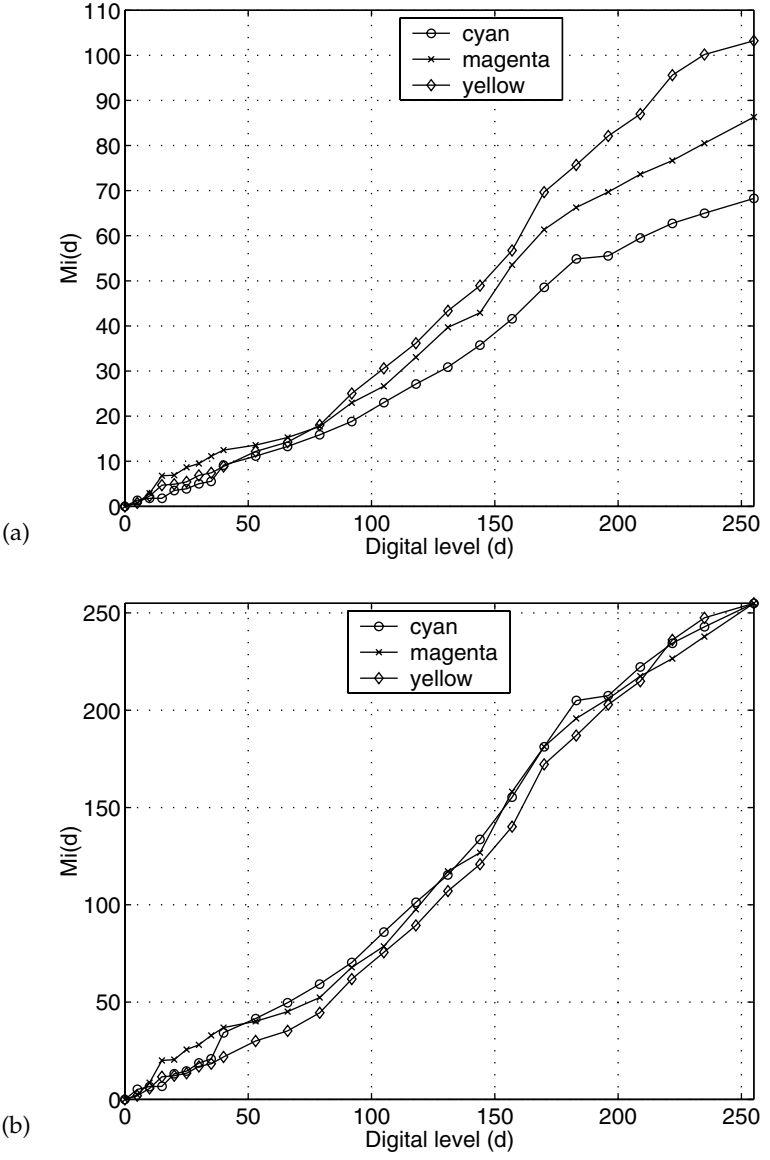


Figure 5.28 Raw device response, $M_i(d)$ defined as ΔE^*_{ab} from paper, for Xerox DocuColor 12 printer: (a) unscaled and (b) scaled to d_{max}.

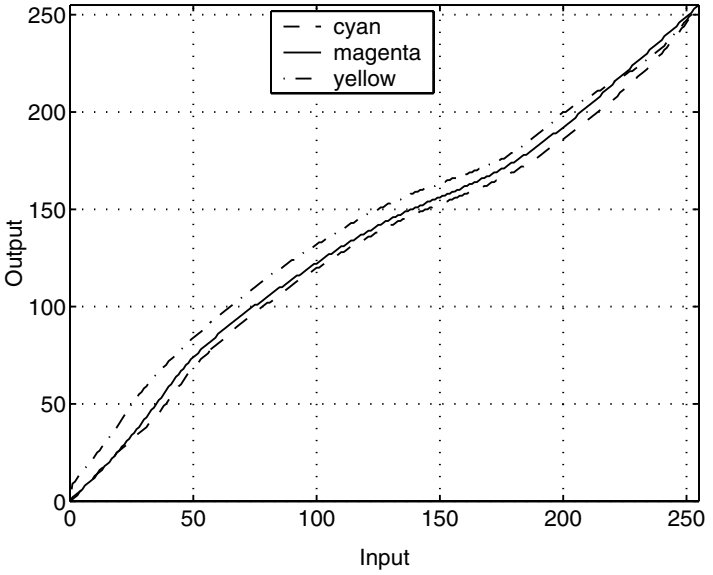


Figure 5.29 Calibration curves correcting for response of Fig 5.28.

calibration, the same C, M, Y stepwedge data were processed through the calibration TRCs, printed, and measured, and M_i was evaluated using Equation 5.57 and plotted in Figure 5.30. The calibrated response is now linear with respect to the desired metric M_i.

Other metrics can be used instead of Equation 5.57, e.g., optical density, or luminance.[48] The calibration procedure is identical.

5.10.1.2 Gray-balanced calibration

An alternative approach to calibration is to gray balance the printer so that equal amounts of C, M, Y processed through the calibration result in a neutral (i.e., $a^* = b^* = 0$) response. There are two main motivations for this approach. First, the human visual system is particularly sensitive to color differences near neutrals; hence, it makes sense to carefully control the state of the printer in this region. Second, gray balancing considers, to a first order, interactions between C, M, and Y that are not taken into account in channel-independent calibration. However, gray balancing is more complicated than channel-independent linearization and generally demands a larger number of patch measurements.

In addition to determining the relative proportions of C, M, Y that generate neutral colors, gray balancing can also achieve a specified tone response along the neutral axis (e.g., linear in neutral luminance or lightness). The following procedure can be used to gray balance and linearize the printer to neutral lightness L^*:

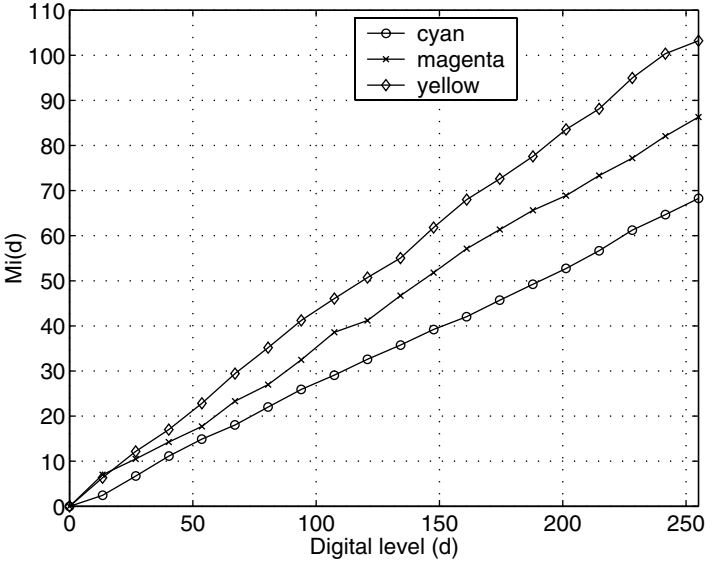


Figure 5.30 Response $M_i(d)$ of calibrated device.

1. Generate a training set of device-dependent (CMY) data in the vicinity of neutrals across the dynamic range of the printer. One exemplary approach is to vary C and M for fixed Y, repeating for different levels of Y across the printer range. The number and spacing of steps for C, M, and Y should be chosen so as to bracket the neutral $a^* = b^* = 0$ axis. Therefore, these parameters depend on printer characteristics, and their selection will require some trial and error.
2. Generate device-independent (CIELAB) data corresponding to these patches. This can be accomplished either via direct measurement of a target containing these patches or by processing the CMY data through a printer model that predicts the colorimetric response of the printer. (Printer models are discussed in a subsequent section.) Media-relative colorimetry is recommended for the CIELAB calculations. If the CIELAB measurements do not bracket the neutral axis, it may be necessary to iterate between this and the previous step, refining the choice of CMY points at a given iteration based on the CIELAB measurements from the previous iteration.
3. Given the training CMY and CIELAB data, obtain CMY values that yield neutral measurements, i.e., $a^* = b^* = 0$, at a set of lightness levels L_i^*, $i = 1, \ldots, T$, spanning the range of the printer. A sufficiently fine sampling of measurements may allow the neutral points to be directly selected; however, in all likelihood, some form of fitting or interpolation will be required to estimate neutral points. A possible candidate is the distance-weighted linear regression function from

Section 5.4.4.2. The regression is supplied with the training data as well as a set of input neutral CIELAB points (L_i^*, 0, 0). The output from the regression is a set of weighted least-squares estimates (C_i, M_i, Y_i) that would produce (L_i^*, 0, 0). A hypothetical example is shown in Figure 5.31a. Typically, 6 to 10 L^* levels are sufficient to determine gray-balance throughout the printer's dynamic range.

4. To generate a monotonically increasing calibration function from Figure 5.31a, invert the sense of the lightness values L_i^* to obtain neutral "darkness" values, denoted D_i^*, scaled to the maximum digital count d_{max}. The formula is given by

$$D_i^* = \left(\frac{d_{\max}}{100}\right)(100 - L_i^*) \tag{5.58}$$

5. Group the data into three sets of pairs $\{D_i^*, C_i\}$, $\{D_i^*, M_i\}$, $\{D_i^*, Y_i\}$, and from this generate three functions, *C(D*), M(D*), Y(D*)* using a one-dimensional fitting or interpolation algorithm. The use of splines is recommended, as these have the flexibility to fit data from a wide variety of printers and also possess the ability to smooth out noise in the data. These functions are plotted in Figure 5.31b for the same hypothetical example. Note that, above a certain darkness $D_{maxgray}^*$, it is not possible to achieve neutral colors, because one of the colorants (cyan in this example) has reached its maximum digital value. Hence, there are no real calibration data points in the input domain $[D_{maxgray}^*, d_{max}]$. One approach to complete the functions is to pad the calibration data with extra values in this region so that the spline

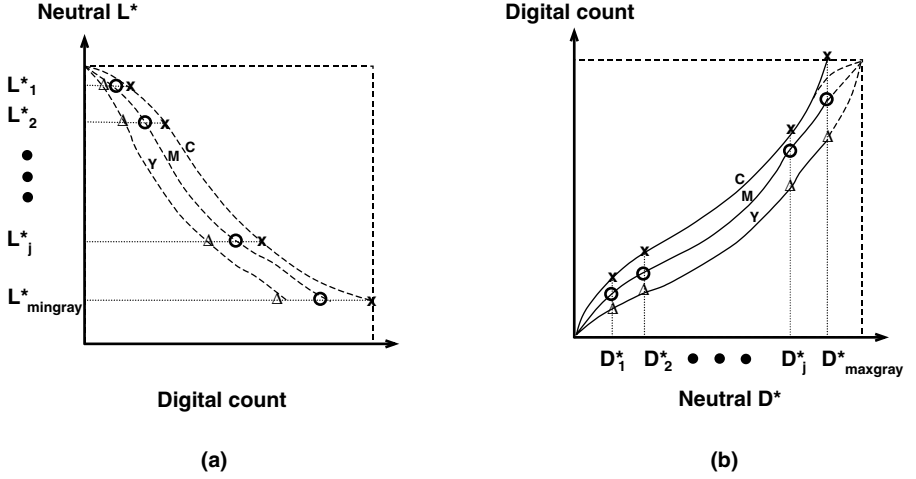


Figure 5.31 Illustration of gray-balance calibration for printers: (a) L^* vs. digital count for neutral samples, and (b) corresponding TRC.

fitting will smoothly extrapolate to the endpoint, d_{max}. Figure 5.31b shows schematically the extrapolation with dashed lines. To achieve smooth calibration functions, it may be necessary to sacrifice gray balance for some darkness values less than $D^*_{maxgray}$. In the case of CMYK printers, the trade-off can be somewhat mitigated by using the K channel in combination with C, M, Y to achieve gray balance. Trade-offs between smoothness and colorimetric accuracy are frequently encountered in printer calibration and characterization. Unfortunately, there is no universal solution to such issues; instead, knowledge of the particular printer and user requirements is used to heuristically guide the trade-offs.

6. Test the calibration by processing a stepwedge of samples C = M = Y = d through the TRCs, and printing and measuring CIELAB values. As before, it is convenient to assess the outcome by plotting L^*, a^*, b^* as a function of the input digital count d. For most of the range $0 \le d \le D^*$, the calibration should yield a linear response with respect to L^*, and $a^*, b^* \approx 0$. If the deviation from this ideal aim is within the inherent variability of the system (e.g., the stability and uniformity of the printer), the calibration is of satisfactory accuracy. Recall that, for gray levels darker than D^*_{max}, a linear gray-balanced response is no longer achievable; instead, the printer response should smoothly approach the color of the CMY solid overprint.
7. An additional test, highly recommended for gray-balance calibration, is to generate a target of continuous ramps of C = M = Y, process through the calibration, print, and visually inspect the output to ensure a smooth neutral response. The prints must be viewed under the same illuminant used for the CIELAB calculations in the calibration.

A recent study by the author has shown that visual tolerance for gray in reflection prints is not symmetric about the $a^* = b^* = 0$ point.[49] In fact, people's memory of and preference for gray occurs in the quadrant corresponding to $a^* < 0$, $b^* < 0$. In regions corresponding to positive a* or b*, a dominant hue is more readily perceived; hence, tolerances for gray reproduction in these regions are small. Colloquially phrased, people prefer "cooler" (bluish/greenish) grays to "warmer" (reddish/yellowish) grays. This observation can exploited to improve the robustness of gray balancing for printers. The device could be balanced toward preferred gray (a*, b* < 0) rather than colorimetric gray (a* = b* = 0) with the same procedure described above. The expected advantage is that, by setting the calibration aim-point in a region with large visual tolerance, the errors inevitably introduced by the calibration are less likely to be visually objectionable.

5.10.2 Model-based printer characterization

Several physics-based models have been developed to predict the colorimetric response of a printer. Some of the common models will be described in

this section. Some overlap exists between this section and an earlier chapter on the physics of color. That chapter focuses on modeling the interaction between light, colorants, and medium at the microscopic level. The emphasis here is in modeling the printer at a macroscopic level, with the goal of deriving the forward characterization mapping from device colorant values to device-independent coordinates such as spectral reflectance or CIEXYZ. Derivation of the inverse characterization function is generally independent of the forward model and is thus discussed in a separate section.

To set a framework for the models to follow, it is instructive to examine different ways in which light passes through a uniform colorant layer. These are depicted in Figure 5.32. In Figure 5.32a, light passes though the colorant layer in only one direction. Some of the light is absorbed, and the remaining is transmitted. The absorption and transmission are functions of wavelength, hence the perception that the layer is colored. This layer is said to be transparent and occurs when the colorant particles are completely dissolved in the medium. The dyes in a dye-diffusion printer can be reasonably well approximated by this model. In Figure 5.32b, some of the light is transmitted and some absorbed as in Figure 5.32a. However, due to the presence of discrete particles, some of the light is also scattered. This layer is said to be translucent. Xerographic and certain inkjet printers subscribe to this model. In Figure 5.32c, a much higher presence of discrete particles results in all of the light being either absorbed or scattered. This layer is said to be opaque, and it applies to paints and some inkjet processes. In all cases, transmission, absorption, and scattering are functions of wavelength. We will see shortly that models for predicting the color of uniform colorant layers are based on one of these three scenarios. More details are given in Chapter 3, which focuses on the physics of color.

In the ensuing discussions, we will assume the exemplary case of a three-colorant (CMY) printer. Extension to an arbitrary number of colorants is usually straightforward.

5.10.2.1 Beer–Bouguer model

The Beer–Bouguer (BB) model plays an important role in colorant formulation, being frequently used to predict light transmission through colored

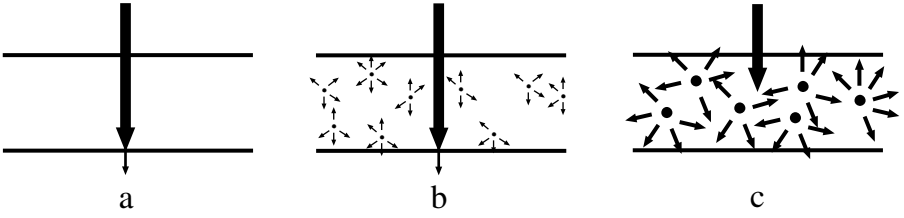


Figure 5.32 Light transport models for (a) transparent, (b) translucent, and (c) opaque media.

materials in liquid solutions. In digital color imaging, it is most applicable for continuous-tone printing with transparent colorants and media (i.e., Figure 5.32a). The underlying assumption is that the spatial rate of change of light radiance as it passes through an absorbing colorant layer is proportional to the radiance itself. Mathematically, this is given by

$$\frac{dI_x(\lambda)}{dx} = -A(\lambda)I_x(\lambda) \tag{5.59}$$

where $I_x(\lambda)$ = radiance at position x within the colorant layer
$A(\lambda)$ = a proportionality factor given by

$$A(\lambda) = \xi\, w\, k(\lambda) \tag{5.60}$$

where ξ = concentration of the colorant
w = thickness of the colorant layer
$k(\lambda)$ = spectral absorption coefficient of the colorant

Substituting Equation 5.60 into Equation 5.59 and integrating with respect to x over the thickness of the colorant, we obtain the following expression for the radiance $I(\lambda)$ emerging from the colorant:

$$I(\lambda) = I_0(\lambda)\exp(-\xi w k(\lambda)) = I_i(\lambda)T_t(\lambda)\exp(-\xi w k(\lambda)) \tag{5.61}$$

where $I_0(\lambda)$ is the radiance of light that would be transmitted in the absence of the colorant, which can be expressed as the product of the incident light $I_i(\lambda)$ and the bare transparency $T_t(\lambda)$. Equation 5.61 essentially states that the amount of light absorption depends directly on the amount of absorbing material within the colorant, which in turn is proportional to both the concentration and thickness of the colorant layer. Often, the colorant thickness w is assumed to be spatially constant and is folded into the absorption coefficient $k(\lambda)$. To this end, we no longer explicitly include w in the analysis.

It is useful to introduce spectral transmittance $T(\lambda)$ and optical density $D(\lambda)$ of a colorant layer.

$$T(\lambda) = \frac{I(\lambda)}{I_i(\lambda)} = T_t(\lambda)\exp(-\xi k(\lambda));$$

$$D(\lambda) = -\log_{10}(T(\lambda)) = D_t(\lambda) + 0.4343\ \xi\ k(\lambda) \tag{5.62}$$

For color mixtures, the additivity rule can be invoked, which states that the density of a colorant mixture is equal to the sum of the densities of the individual colorants.[50] For a CMY printer, we thus have

$$D_{CMY}(\lambda) = D_t(\lambda) + 0.4343(\xi_C k_C(\lambda) + \xi_M k_M(\lambda) + \xi_Y k_Y(\lambda)) \tag{5.63}$$

which can be written in terms of transmittance,

$$T_{CMY}(\lambda) = T_t(\lambda) \exp[-(\xi_C k_C(\lambda) + \xi_M k_M(\lambda) + \xi_Y k_Y(\lambda))] \tag{5.64}$$

The model can be extended to reflective prints under the assumption that there is no scattering of light within the paper. This yields

$$R_{CMY}(\lambda) = R_p(\lambda) \exp[-(\xi_C k_C(\lambda) + \xi_M k_M(\lambda) + \xi_Y k_Y(\lambda))] \tag{5.65}$$

where $R_{CMY}(\lambda)$ and $R_p(\lambda)$ are the spectral reflectances of the colorant mixture and paper, respectively. Note that, in reality, most reflective media do exhibit scattering, hence reducing the accuracy of Equation 5.65.

For a given printer, the parameters of Equations 5.64 and 5.65 are estimated from measurements of selected patches. The procedure for a CMY reflection printer is as follows:

- Measure the spectral reflectance of the paper, $R_p(\lambda)$, and solid C, M, Y patches, $R_C(\lambda)$, $R_M(\lambda)$, $R_Y(\lambda)$.
- Estimate the spectral absorption coefficient $k_C(\lambda)$ for the cyan colorant. This is done by setting $\xi_C = 1$, $\xi_M = \xi_Y = 0$ in Equation 5.65 to yield

$$k_C(\lambda) = -\log\left(\frac{R_C(\lambda)}{R_p(\lambda)}\right) \tag{5.66}$$

  Use analogous expressions to derive $k_M(\lambda)$ and $k_Y(\lambda)$.

- Derive the relationship between input digital level d_j and cyan concentration ξ_{Cj} by printing a stepwedge of pure cyan patches at different digital levels d_j, and measure spectral reflectances $R_{Cj}(\lambda)$. From Equation 5.65, we know that

$$-\log\left(\frac{R_{Cj}(\lambda)}{R_p(\lambda)}\right) = \xi_{Cj} k_C(\lambda) \tag{5.67}$$

  The quantity on the left is the absorption corresponding to concentration at level d_j; hence, we denote this as $k_{Cj}(\lambda)$. A least-squares estimate for ξ_{Cj} can be computed by minimizing the error.

$$\sum_{\lambda} \left\| k_{Cj}(\lambda) - \xi_{Cj} k_C(\lambda) \right\|^2 \tag{5.68}$$

  where the summation is overall measured wavelengths within the visible spectrum. Using the least-squares analysis in Appendix 5.A, the optimal ξ_{Cj} is given by

$$\xi_{Cj}^{opt} = \frac{\sum_{\lambda} k_{Cj}(\lambda) k_C(\lambda)}{\sum_{\lambda} k_C^2(\lambda)} \tag{5.69}$$

By definition, these estimates lie between 0 and 1. Using Equation 5.69, we obtain a set of pairs (d_j, ξ_{Cj}), from which one-dimensional fitting or interpolation is used to generate a TRC that maps digital count to dye concentration for all digital inputs $0 \le d \le d_{max}$. This process is repeated for magenta and yellow.

This completes the model derivation process, and all the parameters in Equation 5.65 are known. The model can be tested by exercising it with an independent set of CMY test data. The model predictions are compared with actual measurements using a standard error metric such as ΔE_{94}. For efficiency of computation, the model can be used to create a three-dimensional LUT that maps CMY directly to CIE coordinates.

The BB model works very well for photographic transparencies and, to a reasonable extent, for photographic reflection prints. One of the shortcomings of this model is that it does not account for scattering within colorant layers, thus reducing its applicability for certain printing technologies. The scattering phenomenon is explicitly introduced in the Kubelka–Munk model, described next.

5.10.2.2 Kubelka–Munk model

The Kubelka–Munk (KM) model is a general theory for predicting the reflectance of translucent colorants. An appealing aspect of this theory is that it also models transparent and opaque colorants as special cases. The foremost applicability for printer characterization is the case of continuous-tone printing processes on reflective media. In this section, only the important formulae are presented. Their derivations are rather lengthy and can be found in many sources, including Allen.[51]

Kubelka–Munk theory assumes a special case of Figure 5.32b, with light being transmitted or scattered in only two directions: up and down. The most general form of the KM model for translucent colorant layers is given by

$$R(\lambda) = \frac{\frac{R_p(\lambda) - R_\infty(\lambda)}{R_\infty(\lambda)} - R_\infty(\lambda)\left(R_p(\lambda) - \frac{1}{R_\infty(\lambda)}\right)\exp\left[wS(\lambda)\left(\frac{1}{R_\infty(\lambda)} - R_\infty(\lambda)\right)\right]}{R_p(\lambda) - R_\infty(\lambda) - \left(R_p(\lambda) - \frac{1}{R_\infty(\lambda)}\right)\exp\left[wS(\lambda)\left(\frac{1}{R_\infty(\lambda)} - R_\infty(\lambda)\right)\right]} \tag{5.70}$$

where $R(\lambda)$ = the spectral reflectance of the sample
$R_p(\lambda)$ = the reflectance of the paper
w = the thickness of the colorant layer
$K(\lambda)$ and $S(\lambda)$ = absorbing and scattering coefficients, respectively

$R_\infty(\lambda)$ = the reflectance of an infinitely thick sample, given by

$$R_\infty(\lambda) = 1 + \frac{K(\lambda)}{S(\lambda)} - \sqrt{\left(\frac{K(\lambda)}{S(\lambda)}\right)^2 + 2\left(\frac{K(\lambda)}{S(\lambda)}\right)} \tag{5.71}$$

In practice, a sample is "infinitely thick" if any increase in thickness results in a negligible change in reflectance. Equation 5.71 can be inverted to obtain

$$\frac{K(\lambda)}{S(\lambda)} = \frac{(1 - R_\infty(\lambda))^2}{2R_\infty(\lambda)} \tag{5.72}$$

For colorant mixtures, the additivity and proportionality rules can be applied to obtain overall absorbing and scattering coefficients from those of the individual colorants.

$$K(\lambda) = k_p(\lambda) + \sum_{i = C,M,Y} \xi_i k_i(\lambda); \quad S(\lambda) = s_p(\lambda) + \sum_{i = C,M,Y} \xi_i s_i(\lambda) \tag{5.73}$$

where $k_p(\lambda)$ and $s_p(\lambda)$ = the absorption and scattering terms for the paper
ξ_i = the concentration of colorant i

The general KM model, Equation 5.70, can be simplified to the two limiting cases of transparent and opaque colorants (Figure 5.32a and 5.32c), described next.

5.10.2.2.1 KM model for transparent colorants. For transparent colorant layers, the scattering term in Equation 5.70 approaches zero, resulting in the following expression:[51]

$$R(\lambda) = R_p(\lambda)\exp[-2wK(\lambda)] \tag{5.74}$$

where $K(\lambda)$ is given by Equation 5.73. Note that this is very similar to the Beer–Bouguer model, Equation 5.65. However, the absorption coefficients in the two models are different, because BB assumes collimated light, whereas KM assumes diffuse light. The procedure outlined in Section 5.10.2.1 for the BB model can be used to estimate $k_i(\lambda)$ from C, M, Y samples at maximum concentration and to derive the mapping between input digital value d_i and dye concentration ξ_i from stepwedge measurements.

Berns has used this model to characterize dye diffusion printers.[52] In this work, the model parameters [i.e., $k_i(\lambda)$ and ξ_i, i = C, M, Y] were initially derived using essentially the procedure outlined in Section 5.10.2.1. A third-order polynomial was used to fit the relationship between digital count and dye concentration. The model resulted in unsatisfactory results (ΔE_{ab} = 12).

It was discovered that a major source of error arose from the channel independence assumption in the KM model, i.e., the cyan dye concentration depends only on the cyan digital count, etc. The author observed that, due to the sequential nature of the dye diffusion and transfer process, there is a significant sequential interaction among the colorants. This was accounted for by introducing a matrix with cross terms to relate KM predictions to more realistic estimates. Coefficients of the matrix were obtained by regression on a set of measurements of colorant mixtures. This correction was found to significantly improve the model prediction, resulting in $\Delta E_{ab} = 3$. Details are given in the Berns reference. The empirical correction just described is a common way of accounting for limitations in a physics-based model and will be encountered again in discussions of the Neugebauer model.

5.10.2.2.2 KM model for opaque colorants. For opaque samples, the limiting case of infinite thickness in Equation 5.71 can be used to predict spectral reflectance. Note that Equation 5.71 depends only on the ratio $K(\lambda)/S(\lambda)$ for the colorant mixture, which can be obtained from the absorption and scattering coefficients of the individual colorants using Equation 5.73.

$$\frac{K(\lambda)}{S(\lambda)} = \frac{k_p(\lambda) + \sum_{i=C,M,Y} \xi_i k_i(\lambda)}{s_p(\lambda) + \sum_{i=C,M,Y} \xi_i s_i(\lambda)} \tag{5.75}$$

This is referred to as the two-constant KM model. With certain pigments, it is reasonable to assume that the scattering in the colorants is negligible compared to scattering in the substrate.[51] In this case, the denominator in Equation 5.75 reduces to $s_p(\lambda)$, and Equation 5.75 can be rewritten as

$$\frac{K(\lambda)}{S(\lambda)} = \frac{k_p(\lambda)}{s_p(\lambda)} + \sum_{i=C,M,Y} \xi_i \frac{k_i(\lambda)}{s_i(\lambda)} \tag{5.76}$$

This is referred to as the single-constant KM model, as only a single ratio $k(\lambda)/s(\lambda)$ is needed for each colorant.

To derive the model, the $k(\lambda)/s(\lambda)$ terms for each colorant are obtained from reflectance measurements of samples printed at maximum concentration, using Equation 5.72. Next, the relationship between digital count and colorant concentration ξ are obtained from reflectance measurements of single-colorant stepwedges and a regression procedure similar to that outlined in Section 5.10.2.1. Finally, Equations 5.76 and 5.71 are evaluated in turn to obtain the predicted reflectance. More details are found in papers by Parton et al.[53] and Kang.[54] In these papers, the opaque single-constant KM model is used to predict the spectral reflectance of solid area coverage in inkjet prints. The prediction accuracies are in the range of 1.65 to 5.7 ΔE^*_{ab}

depending on the ink mixing process and the particular mixtures tested. Note that most inkjet printers use halftoning, a process that is not well predicted by KM theory. The latter only predicts the solid overprints in inkjet prints, hence its application is in ink formulation rather than device characterization.

5.10.2.2.3 Modeling front-surface and interlayer reflections. An important effect not taken into account in the KM and BB models is reflection loss at the boundaries between colorant layers, as well as front surface reflection (FSR) at the boundary between the uppermost colorant layer and air. Because a certain amount of light is lost due to FSR, this should ideally be subtracted before computing reflectance. However, in a spectrophotometer, at least part of the light from FSR reaches the detector. To correct for this effect, Saunderson[55] developed a relationship between the reflectance $R(\lambda)$ as predicted by BB or KM, and the reflectance $R_{meas}(\lambda)$ as measured by a spectrophotometer.

$$R(\lambda) = \frac{R_{meas}(\lambda) - k_1}{1 - k_1 - k_2(1 - R_{meas}(\lambda))}; \; R_{meas}(\lambda) = k_1 + \frac{(1 - k_1)(1 - k_2)R(\lambda)}{1 - k_2 R(\lambda)} \tag{5.77}$$

where k_1 is the Fresnel reflection coefficient that accounts for front surface reflection, and k_2 models total internal reflection that traps light within the colorant layers. The factor k_1 depends on the refractive index η of the uppermost colorant layer. A common assumption for η is 1.5, which corresponds to $k_1 = 0.04$. The theoretical value of k_2 for the case of perfectly diffuse light is 0.6.[51] Alternatively, these parameters can be chosen to provide the best empirical fit between measured and modeled reflectance data.

The Saunderson correction is performed as a final step after deriving the BB or KM, and it has been shown to improve model accuracy.[52,54]

5.10.2.2.4 Modeling fluorescence. Another drawback with both the BB and KM models is that they do not account for fluorescence. Many paper substrates employ optical brighteners that exhibit fluorescence and can thus limit the utility of these models. Fluorescence modeling is discussed in more detail in Chapter 3, which deals with the physics of color.

5.10.2.3 Neugebauer model

The Neugebauer model is used to model a halftone color printing process. Each primary colorant in a halftone process is rendered as a spatial pattern of dots, each dot being printed at one of a small number of concentration levels. The impression of intermediate levels is achieved by modulating the size, shape, and spatial frequency of the dots. (Techniques for color halftoning are covered in more detail in a subsequent chapter.)

A process employing N colorants at Q concentration levels results in one of Q^N colorant combinations being printed at any given spatial location. We begin the formulation with the simplest case of a binary black-and-white

printer. This corresponds to $N = 1$ and $Q = 2$ (zero or maximum) concentration levels; thus, at any given spatial location, we have two possible colorant combinations, black or white. The reflectance of a halftone pattern is predicted by the Murray–Davies equation,[50]

$$R = (1 - k)P_p + kP_k \tag{5.78}$$

where k = fractional area covered by the black dots
P_p, P_k = reflectances of paper and black colorant, respectively

The Neugebauer model is a straightforward extension of the Murray–Davies equation to color halftone mixtures.[56] Binary printers employing C, M, Y colorants render one of $2^3 = 8$ colorant combinations at a given spatial location. The set of colorant combinations is S = {P, C, M, Y, CM, MY, CY, CMY}, where P denotes paper white, C denotes solid cyan, CM denotes the cyan–magenta overprint, etc. The Neugebauer model predicts the reflectance of a color halftone as a weighted average of the reflectances of the eight colorant combinations.

$$R = \sum_{i=S} w_i P_i \tag{5.79}$$

where S = the aforementioned set of colorant combinations
P_i = spectral reflectance of the ith colorant combination, henceforth referred to as the ith Neugebauer primary
weight w_i = the relative area coverage of the ith colorant combination, which is dictated by the halftoning method used

In the original Neugebauer equations, the predicted color is specified by three broadband reflectances representing the short, medium, and long wavelength portions of the electromagnetic spectrum. In this work, spectrally narrowband reflectances are used instead of their broadband counterparts, as the former generally yield greater accuracy.[57] The spectral Neugebauer equations are

$$R(\lambda) = \sum_{i \in S} w_i P_i(\lambda) \tag{5.80}$$

Because $P_i(\lambda)$ are colors of solid overprints, they can be predicted from single-colorant measurements using the BB or KM theories described in the previous sections. However, for any given set of colorants, there are only a small number of such overprints; hence, they are usually measured directly.

5.10.2.3.1 Effect of halftone dot placement. A common assumption is that the dot placements of the colorants are statistically independent; i.e.,

the event that a particular colorant is placed at a particular spatial location is independent of other colorants being placed at the same location. This leads to the Demichel dot model.[50] The Neugebauer primaries and the corresponding weights are given by

$$\begin{aligned} P_i(\lambda) \in S_P &= \{P_P(\lambda), P_C(\lambda), P_M\lambda, P_Y(\lambda), P_{CM}(\lambda), P_{CY}(\lambda), P_{MY}(\lambda), P_{CMY}(\lambda)\}, \\ w_i \in S_w &= \{(1-c)(1-m)(1-y), c(1-m)(1-y), m(1-c)(1-y), \\ &\quad y(1-c)(1-m), cm(1-y), cy(1-m), my(1-c), cmy\} \end{aligned} \tag{5.81}$$

Here, c, m, y are the fractional area coverages corresponding to digital inputs d_c, d_m, d_y, respectively. A halftone screen for which statistical independence is often assumed is the rotated halftone screen configuration, where the screens for c, m, y are placed at different angles, carefully selected to avoid moiré artifacts. This is shown schematically in Figure 5.33a. Validity of the independence assumption for certain types of halftones such as rotated screens has been demonstrated by Viggiano et al.[58]

A geometrical interpretation of Equations 5.80 and 5.81 is that $R(\lambda)$ is a result of trilinear interpolation performed among the $P_i(\lambda)$ in cmy space. (This can be verified by comparing these equations with the trilinear interpolation equations given in Chapter 11, dealing with efficient color transformations.) An algebraic interpretation of the model is that Equations 5.80 and 5.81 form a third-order polynomial in terms of c, m, y, with $P_i(\lambda)$ being the polynomial coefficients.

Another commonly used halftone configuration is the dot-on-dot screen,[59] where the C, M, Y dots are placed at the same screen angle and

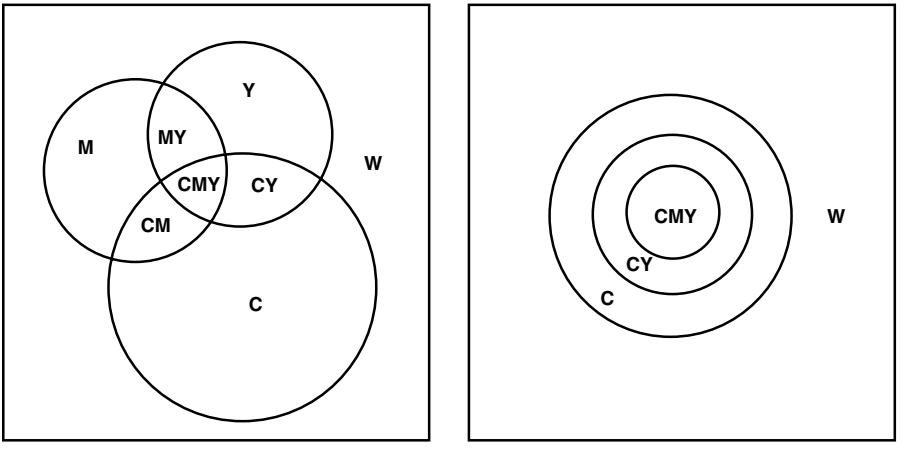


Figure 5.33 Dot area coverages for (a) rotated screen and (b) dot-on-dot screen.

phase as shown in Figure 5.33b. While the basic form of the mixing equations is similar to Equation 5.80, the weights w_i are different from those of randomly positioned dots. Let X_i be the colorant with the ith smallest area coverage a_i. For example, if [c, m, y] = [0.8, 0.5, 0.2], then X_1 = Y, X_2 = M, X_3 = C; $a_1 = 0.2$, $a_2 = 0.5$, $a_3 = 0.8$. The set of Neugebauer primaries and corresponding weights are now given by

$$P_i(\lambda) \in S_P = \{P_{X_1X_2X_3}(\lambda),\ P_{X_2X_3}(\lambda),\ P_{X_3}(\lambda),\ P_w(\lambda)\},$$
$$w_i \in S_w = \{a_1, a_2 - a_1, a_3 - a_2, 1 - a_3\} \tag{5.82}$$

The final output reflectance in Equation 5.80 is now a summation of, at most, four terms.

Geometrically, Equation 5.82 represents tetrahedral interpolation among the $P_i(\lambda)$ at four of the eight vertices of the *cmy* cube. (This can be verified by comparing Equation 5.82 with the equations for tetrahedral interpolation given in the Chapter 11.) Different Neugebauer primaries are selected for the calculation depending on the relative sizes of the area coverages *c, m, y* (equivalently, the tetrahedron to which the input *cmy* coordinate belongs). However, the weights w_i, and hence the resulting interpolated output $R(\lambda)$, are continuous as the input *cmy* coordinate moves from one tetrahedron to another in *cmy* space. Algebraically, it is easily seen from Equations 5.80 and 5.82 that, for fixed λ, $R(\lambda)$ is a linear function of *c, m, y*, with the $P_i(\lambda)$ being the weighting coefficients.

The aforementioned dot-on-dot mixing model assumes an ideal dot pattern with no noise, a perfectly rectangular dot density profile, and no misregistration effects. In practice, these assumptions may be violated. It has been shown[59] that a weighted mixture of the dot-on-dot and Demichel mixing models can effectively capture some of these effects. The new predicted reflectance is given by

$$R'(\lambda) = (1 - \alpha)R_{dod}(\lambda) + \alpha R_{dem}(\lambda) \tag{5.83}$$

where $R_{dod}(\lambda)$ = reflectance predicted by the dot-on-dot model
$R_{dem}(\lambda)$ = reflectance predicted by the Demichel model
α = a weighting parameter that determines the relative proportions of the two mixing models; this factor can be chosen to fit the model to a set of measured data

As alluded to earlier, all these versions of the Neugebauer model easily generalize for an arbitrary number of colorants. For N colorants, the Demichel model for independent dot placement will result in the summation in Equation 5.80 containing 2^N terms, while the dot-on-dot model contains $N + 1$ terms.

5.10.2.3.2 Effect of halftone screen frequency. The ideal Neugebauer model assumes a perfect rectangular dot profile as a function of spatial location. In reality, dots have soft transitions from regions with full colorant to regions with no colorant. If the halftone screen frequency is relatively low, or a clustered dot is used, the relative area of the paper covered by the transition regions is small, and the Neugebauer model would be expected to be relatively accurate. On the other hand, if the screen frequency is high, or a dispersed dot is used, a relatively large fraction of the paper is covered by transition regions, and the model breaks down. While some of the corrections discussed in following sections partially account for soft transitions, the reliability of the model has been seen to be greatest with clustered dot screens with frequency less than 100 halftone dots per inch.

5.10.2.3.3 Effect of light scattering in the paper. An important phenomenon not modeled by the basic Neugebauer equations is the scattering of light within the paper. To understand this phenomenon, consider the interaction of light with a black halftone print. The light that reaches the paper is given by

$$I_p = I_{in}(1 - k + kT_k) \tag{5.84}$$

where I_{in} = incident light intensity
I_p = light reaching the paper
k = fractional black area coverage
T_k = transmittance of the black colorant

Figure 5. 34a shows the case where there is no optical scattering within the paper. In this case, light incident on the print at a location containing colorant will also exit through the colorant; likewise, light reaching the substrate will exit from the same location. The reflected light is thus given by

$$I_{refl} = I_{in}\{(1 - k)\ P_p + k\ T_k^2\ P_p\} \tag{5.85}$$

where P_p is reflectance of the paper.

Define the reflectance of the solid black colorant as $P_k = T_k^2 P_p$. The overall reflectance is then given by the original Murray–Davies Equation 5.78.

Consider now the case where there is scattering within the paper, as shown in Figure 5.34b. In this case, light that enters the paper through an area with no colorant may leave the paper through an area that is covered with colorant, and vice versa. To account for this, Yule and Nielsen[60] proposed a simple correction to the Murray–Davies model for a black printer. Assuming that light reaching the paper is given by I_p in Equation 5.84, the light emerging from the substrate is $I_p R_p$. If complete diffuse scattering is assumed, the light is equally likely to re-emerge from the paper in all directions. In this case,

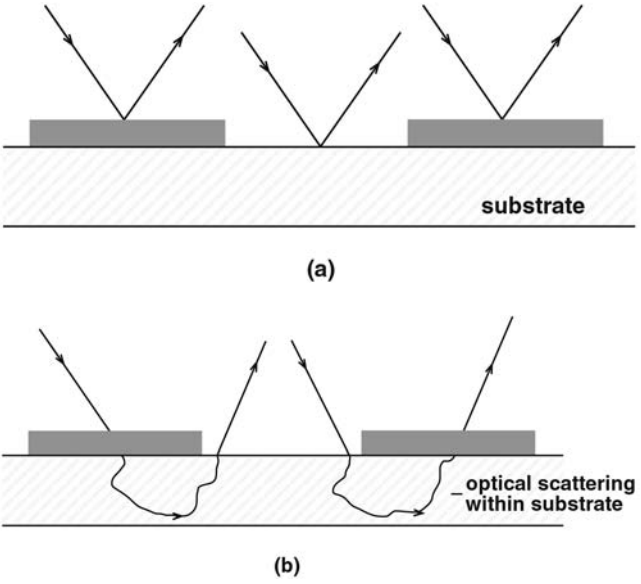


Figure 5.34 Light reflection (a) without and (b) with optical scattering within the substrate.

the emerging light experiences the same transmission function, $(1 - k + kT_k)$, as in Equation 5.84. The final reflected light is given by

$$I_{reft} = I_p P_p (1 - k + kT_K) = I_{in} P_p (1 - k + kT_K)^2 \tag{5.86}$$

With the black reflectance being defined as $P_k = T_k^2 P_p$, the following expression is obtained for the overall reflectance:

$$R = \frac{I_{reft}}{I_{in}} = ((1 - k)P_p^{1/2} + kP_k^{1/2})^2 \tag{5.87}$$

The Yule–Nielsen (YN) correction results in a nonlinear relationship between the area coverage k and the resulting reflectance R. Figure 5.35 is a plot of R vs. k with and without the YN correction. The latter predicts a smaller reflectance (i.e., a darker print) than the linear Murray–Davies model. This is indeed the case in reality. The darker print can be thought of as being effected by a larger dot area coverage k; hence, the scattering phenomenon is often referred to as *optical dot gain.*

Equation 5.87 can be generalized as follows:

$$R = ((1 - k)P_p^{1/n} + kP_k^{1/n})^n \tag{5.88}$$

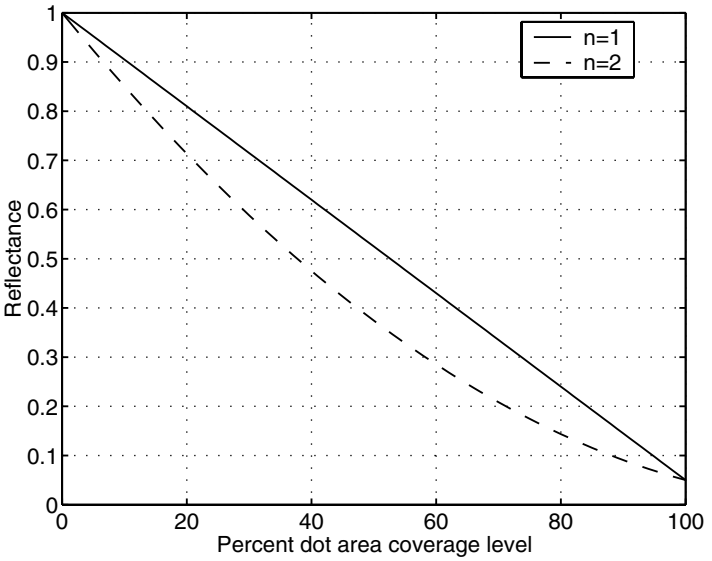


Figure 5.35 Reflectance vs. area coverage for *K* colorant (a) without Yule–Nielsen correction ($n = 1$) and (b) with Yule–Nielsen correction ($n = 2$).

where n is known as the YN parameter. When $n = 1$, Equation 5.88 reduces to the Murray–Davies equation, i.e., the case of no optical scattering within the substrate. When $n = 2$, we have the case of complete diffuse optical scattering given in Equation 5.87. In reality, one would expect to encounter partial diffuse scattering, which would yield intermediate values, $1 < n < 2$. Therefore, n is often treated as a free parameter chosen to optimally fit measured data. The YN correction is readily applied to the spectral Neugebauer equations.

$$R(\lambda) = \left(\sum_{i \in S} w_i P_i(\lambda)^{1/n} \right)^n \tag{5.89}$$

Figure 5.36 is a plot of the prediction accuracy of the Neugebauer model as a function of n for a rotated screen. The device being modeled was a Xerox 5760 CMYK laser printer. Details of the experiment that produced these results are given in the paper by Balasubramanian.[59] Clearly, inclusion of the YN factor (i.e., $n > 1$) greatly improves model accuracy. Interestingly, for this case, best results are achieved for $n > 2$, for which there is no direct physical interpretation. Statistical or empirical fitting of model parameters can indeed often result in nonphysical values. This is largely due to noise and other characteristics such as front surface and internal reflections not being sufficiently captured by the given model.

Other, more sophisticated techniques have been proposed that model optical scattering with spatial point spread functions.[61,62] These approaches

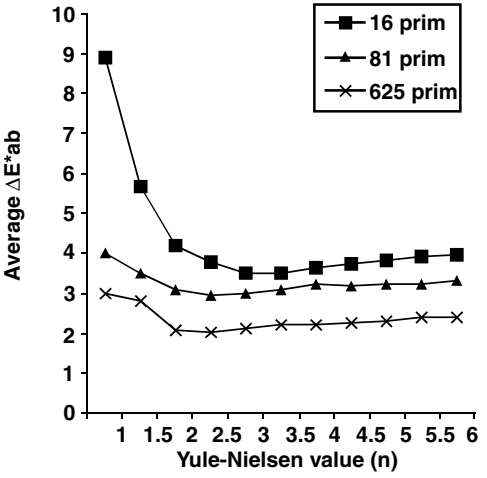


Figure 5.36 Average ΔE vs. YN parameter n for spectral Neugebauer model with $2^4 = 16$, $3^4 = 81$, and $5^4 = 625$ primaries, for rotated dot screen.

are covered in more detail in Chapter 3. The following discussion is restricted to the YN correction, as it is a very simple yet effective way of improving model accuracy.

5.10.2.3.4 Estimation of dot area coverages. In addition to the optical dot gain just described, halftone printing also experiences mechanical dot gain, which results from the physical spreading of colorant on the paper. A combination of optical and mechanical dot gain results in a nonlinear relationship between the input digital counts to the halftone function and the dot area coverages used in the Neugebauer calculation. Furthermore, in some printing processes, optical interactions among the colorants can result in the dot gain for a given colorant being dependent on the area coverages of the other colorants. However, for pedagogical purposes, we will make the simplifying assumption that there are no interchannel interactions so that the cyan area coverage depends on only the cyan digital count, etc. This assumption is reasonably upheld in many printing processes and allows the relationship between digital count and dot area to be determined from single-colorant stepwedge data. From Equation 5.89, the reflectance of a cyan patch produced at digital level d_j is given by

$$R_{C_i}(\lambda)^{1/n} = (1 - c_j)P_p(\lambda)^{1/n} + c_j P_C(\lambda)^{1/n} \tag{5.90}$$

The least-squares estimate minimizes the error

$$E = \sum_{\lambda \in V} [R(\lambda)_{C_j}^{1/n} - ((1 - c_j)P_P(\lambda)^{1/n} + c_j P_C(\lambda)^{1/n})]^2 \tag{5.91}$$

The optimal area coverage is obtained by setting to zero the partial derivative of Equation 5.91 with respect to c_j, yielding

$$c_j^{opt} = \frac{\sum_{\lambda}(P_p(\lambda)^{1/n} - R(\lambda_{c_j})^{1/n})(P_P(\lambda)^{1/n} - P_C(\lambda)^{1/n})}{\sum_{\lambda}(P_P(\lambda)^{1/n} - P_C(\lambda)^{1/n})^2} \tag{5.92}$$

The result is a set of pairs $\{d_j, c_j\}$ from which a continuous function can be derived that maps digital count to dot area coverage using some form of one-dimensional fitting or interpolation. The process is repeated for the other colorants. If a sufficiently fine sampling of stepwedge data is available, piecewise linear interpolation should be adequate; otherwise, higher-order functions such as splines are desirable. Figure 5.37 shows optimized magenta dot areas for the DocuColor 12 printer for values of $n = 1$ and $n = 2$. For the case where $n = 1$, the dot area coverages must account entirely for both optical and mechanical dot gain. When $n > 1$, the YN correction partially accounts for optical dot gain; hence, the dot area coverages are generally smaller in magnitude.

An alternative technique for determining dot areas is to minimize the error in CIELAB rather than spectral coordinates. Unlike the previous approach, this is a nonlinear optimization problem that must be solved with numerical or search-based techniques. Given this fact, one can extend the training set to include colorant mixtures, e.g., C = M = Y, in addition to the single-colorant stepwedges. Balasubramanian[59] provides further details of this approach.

5.10.2.3.5 Cellular Neugebauer model. The set of primaries $P_i(\lambda)$ of the basic Neugebauer model are derived from C, M, Y overprints of either 0 or 100% area coverages. This set can be generalized to include intermediate

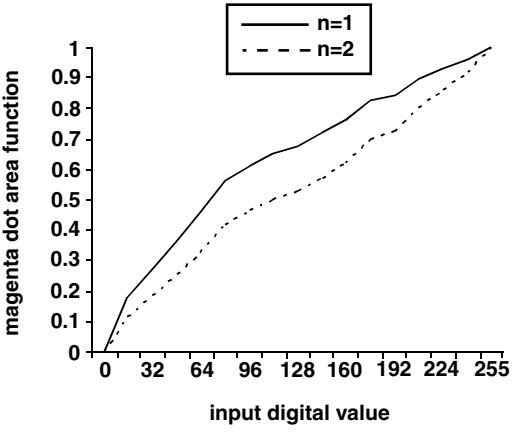


Figure 5.37 Optimized magenta dot area functions for $n = 1$ and 2.

area coverages. For example, if 50% area coverages of C, M, Y are included with 0 and 100%, then each colorant has three states, and there are $3^3 = 27$ Neugebauer primaries. Geometrically, this is equivalent to partitioning the three-dimensional *cmy* space into a grid of eight rectangular cells, formed by nodes at 0, 50, and 100%. Hence, this is referred to as the cellular Neugebauer model.[63] A two-dimensional example is shown in Figure 5.38 for a printer employing only cyan and magenta colorants. Depending on the type of halftone screen, the appropriate mixing equations are applied within each cell. The mixing equations are to be geometrically interpreted as a three-dimensional interpolation of the $P_i(\lambda)^{1/n}$ at the cell vertices. For the case of the random halftone, the logical extension from the noncellular model is to perform trilinear interpolation within each cell whereas, for the dot-on-dot case, tetrahedral interpolation is to be applied.

More explicitly, a given set of dot areas c, m, y can be represented as a point in three-dimensional *cmy* space and will fall in a rectangular cell that is bounded by the lower and upper extrema, denoted c_l, c_u, m_l, m_u, y_l, y_u, along each of the three axes. That is, c_l and c_u are the two points along the cyan axis that satisfy the constraint $0 \le c_l < c < c_u \le 1$; c_l, $c_u \in I_c$, where I_c is the set of allowable states or area coverages for the Neugebauer primaries corresponding to the cyan colorant. Analogous definitions hold for the magenta and yellow coordinates. To estimate the reflectance within a given cell, the dot area values c, m, y, must be normalized to occupy the interval [0, 1] within that cell.

$$c' = \frac{c - c_l}{c_u - c_l} \tag{5.93}$$

with analogous expressions for m' and y'. The weights w_i' for the cellular model are then given by Equation 5.81 for random screens and Equation 5.82 for dot-on-dot screens, with c, m, y being replaced by c', m', y', respectively. Let $P_i'(\lambda)$ be the spectral Neugebauer primaries that correspond to the vertices of the enclosing cell. The mixing equations for the cellular model are then given by Equation 5.89, with w_i replaced by w_i' and P_i replaced by P_i'.

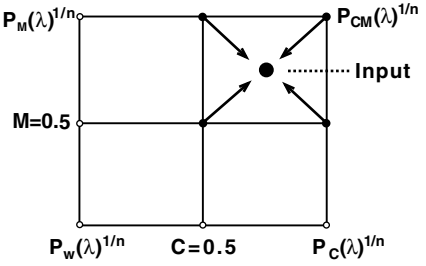


Figure 5.38 Two-dimensional illustration of cellular Neugebauer model. Solid circles denote spectral primaries interpolated to obtain reflectance $R(\lambda)^{1/n}$ at the input *cm* value.

Note that the cellular equations physically model a halftoning process wherein each colorant can produce $M > 2$ concentration levels. For binary printers, the justification for using a cellular model is empirical rather than physical; the finer cellular subdivision of *cm*y space affords finer interpolation of measured data, hence yielding greater accuracy.

Figure 5.36 compares the accuracy of the noncellular model for a CMYK printer with cellular versions employing $3^4 = 81$ and $5^4 = 625$ primaries. As the number of cells increases, the model accuracy improves significantly. At the same time, the dependence on the YN factor decreases. This is to be expected, as the cellular model marks a transition from a model-based to an empirical approach and hence would be less sensitive to model parameters.

5.10.2.3.6 Spectral regression of the Neugebauer primaries. Thus far, the primaries $P_i(\lambda)$ in Equation 5.89 are considered as fixed parameters that are directly measured. An alternative is to treat these quantities as free variables that can be optimized via regression on a training set of spectral reflectance data. This technique will be described next for the case of a noncellular CMY model employing rotated screens. (Extension to the cellular case, N colorants, or dot-on-dot screen is straightforward). It is assumed that the optimal n factor and dot area coverages have been derived using the aforementioned techniques. To formulate the regression problem, it is convenient to express the Neugebauer equations in matrix-vector form. Consider each spectral measurement as an L-vector. Collect the YN modified spectral reflectances $R(\lambda)^{1/n}$ of T training samples into a $T \times L$ matrix **R**. Similarly, collect the YN modified Neugebauer primaries $P_i(\lambda)^{1/n}$ into an $8 \times L$ matrix **P**. Finally, generate a $T \times 8$ weight matrix **W** whose element w_{ij} is the area coverage of the jth Neugebauer primary for the ith training sample. Equation 5.89 can then be rewritten as

$$\mathbf{R} = \mathbf{W} \cdot \mathbf{P} \tag{5.94}$$

From Appendix 5.A, the least squares solution for **P** is given by

$$\mathbf{P}_{opt} = (\mathbf{W}^t\mathbf{W}^{-1})\mathbf{W}^t\mathbf{R} \tag{5.95}$$

The terms in **P** are raised to the power n to obtain optimized primary reflectances. It must be emphasized that the choice of CMY samples in the training set **T** is crucial in determining the condition or rank of matrix **W**. Namely, to ensure sufficient rank, the samples should be chosen so that there are no null columns in **W**. A simple way to assure this is to pick a regular three-dimensional grid of training samples. Also, note that the foregoing analysis is based on a particular choice of n and the dot area coverage functions. The process can be iteratively repeated by rederiving n and the dot areas corresponding to the newly optimized primaries, and then repeating the regression step. Experiments by the author have shown that more than two iterations do not generally yield significant improvements in model accuracy.[59]

5.10.2.3.7 Overall model optimization. The following procedure may be used to optimize the various parameters of the Neugebauer model for a CMY printer:

- Select the resolution of the cellular Neugebauer model. In the author's experience, three levels (i.e., two cells) per colorant offers an acceptable trade-off between accuracy and number of samples required. Generate CMY combinations corresponding to the cell nodes (i.e., the Neugebauer primaries).
- Select the resolution of C, M, Y stepwedges to generate dot area functions. In the author's experience, a minimum of 16 samples per colorant is usually adequate.
- Select an additional set of CMY mixtures to test and refine the model. One possibility is to use an $N \times N \times N$ grid of CMY combinations that does not coincide with the Neugebauer primaries.
- Combine the above CMY samples into a characterization target. (As an alternative to designing a custom target, the standard IT8.7/3 printer characterization target described in Section 5.3 can be used, as it contains the patches necessary to derive and test the Neugebauer model.) Print the target and obtain spectral measurements.
- For a fixed value of n (e.g., $n = 1$), use Equation 5.92 to generate estimates of dot area coverages for the stepwedge samples. Interpolate or fit the data to create functions that map digital count to dot area coverages. With 16 or more samples per stepwedge, piecewise linear interpolation should produce adequate accuracy.
- Evaluate the accuracy of the model in predicting the stepwedge data. This is accomplished by computing a ΔE metric between model predictions with actual measurements.
- Optimize the model with respect to n by repeating the previous two steps for several n values in some nominal range (e.g., $1 < n < 7$) and selecting the n that produces the minimum ΔE.
- Select a mixing model depending on the type of halftone screen (e.g., Demichel vs. dot-on-dot).
- If the dot-on-dot model is chosen, find the optimal blending parameter α in Equation 5.83 by iterating through different values of α, computing ΔE for the model's prediction of mixed color samples from the test set, and selecting α that minimizes the ΔE.
- If spectral regression of the primaries is desired, select a set of mixed color samples from the test set, and use Equation 5.95 to compute optimal primaries $P_i(\lambda)$.

Figure 5.39 summarizes the steps in the application of the Neugebauer model. Accuracy of the model must be evaluated on an independent set of CMY samples. If the prediction error is within the variability of the printer, the model is considered to be a satisfactory representation of the real printer.

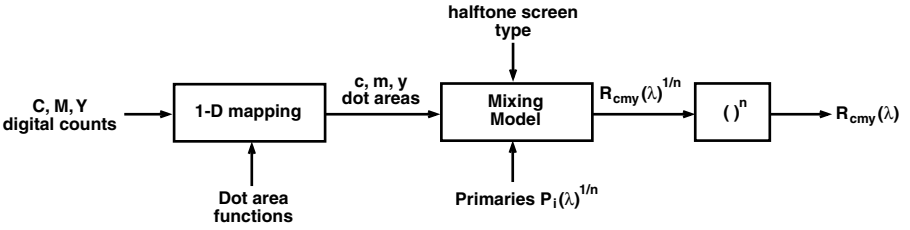


Figure 5.39 Block diagram of Neugebauer model calculation.

5.10.2.3.8 Accuracy of the various Neugebauer models. Table 5.1 compares the performance of the various types of Neugebauer models applied to the Xerox 5760 CMYK printer. Details are provided by Balasubramanian.[59] Clearly, the YN parameter offers significant benefit to the model. The cellular framework with $5^4 = 625$ primaries offers the best accuracy, but this is at the expense of a substantial number of measurements. The cellular model with $3^4 = 81$ primaries, as well as spectral regression, offer a promising trade-off between measurement cost and accuracy.

Table 5.1 Effort Involved and Resulting Accuracy of the Various Neugebauer Models for a Rotated Dot

Model	No. of Spectral Measurements	Avg. ΔE^*_{ab}	95% ΔE^*_{ab}
Basic spectral	72	8.88	16.3
Yule–Nielsen corrected	72	3.50	7.80
Cellular, 3^4 primaries, Yule–Nielsen corrected	137	2.96	6.0
Cellular, 5^4 primaries, Yule–Nielsen corrected	681	2.01	5.0
Yule–Nielsen corrected, global spectral regression	188	2.27	5.3

5.10.2.3.9 Further enhancements. Several researchers have explored other refinements of the model. Arney et al.[64] showed that the colors of both the paper and the dots are functions of the relative dot area coverages, and they extended the Neugebauer model to account for this. Lee et al.[65] departed from the Demichel model and used a sequential quadratic programming method to estimate these parameters. Iino and Berns[66] accounted for optical interactions among the colorants by introducing a correction to the dot gain of a given colorant that depends on the area coverages of the other colorants. Hua and Huang[67] and Iino and Berns[68,69] explored the use of a wavelength-dependent Yule–Nielsen factor. Agar and Allebach[70] developed an iterative technique of selectively increasing the resolution of a cellular model in those

regions where prediction errors are high. Xia et al.[71] used a generalization of least squares, known as total least-squares (TLS) regression to optimize model parameters. Unlike least-squares regression, which assumes uncertainty only in the output space of the function being approximated, total least-squares assumes uncertainty in both the input and output spaces and can provide more robust and realistic estimates. In this regard, TLS has wide applicability in device characterization.

5.10.3 Empirical techniques for forward characterization

With this class of techniques, a target of known device-dependent samples is generated, printed, and measured, and the characterization function is derived via data fitting or interpolation. Linear regression is generally inadequate for printer characterization; any of the more sophisticated nonlinear techniques described in Section 5.4 are applicable.

5.10.3.1 Lattice-based techniques

Perhaps the most common approach is to generate a regular grid of training samples in m-dimensional device space, print and measure these samples, and use a lattice-based technique to interpolate among the measured colorimetric values (see Section 5.4.5). There is an inherent trade-off between the size and distribution of the sample set and the resulting accuracy. This trade-off must be optimized based on the particular printer characteristics and accuracy requirements. Remember that, if the printer has been calibrated, these functions must be incorporated into the image path when generating the target; hence, they will also affect the overall printer characteristics. If, for example, the printer has been calibrated to be linear in ΔE from paper along each of the primary axes (see Section 5.10.1), then uniform spacing of lattice points is a good choice, as these correspond approximately to equal visual steps. The following is a simple procedure to determine a suitable grid size for a CMY printer, assuming it has been either linearized channelwise to ΔE from paper or gray-balanced and linearized to neutral L^*:

- Generate uniformly spaced lattices of size s^3 in CMY space, where $5 \le s \le 10$. Also generate an independent test target of CMY samples. The latter can be generated by invoking a random number generator for each of the digital values d_c, d_m, d_y or by using a regular lattice that is different from any of the training sets.
- Generate targets for both the lattice and the test data, process through the calibration functions, print, and measure CIELAB values.
- From this data, generate a set of three-dimensional LUTs of size s^3 that map CMY to CIELAB space.
- Select a three-dimensional interpolation technique, e.g., trilinear or tetrahedral interpolation. Process the test CMY samples through each of the LUTs to obtain CIELAB estimates. Compute ΔE between estimated and measured CIELAB.

- Plot average and 95th percentile ΔE as a function of s. A logical choice for the lattice size is the smallest s for which an increase in lattice size does not yield appreciable reduction in 95th percentile ΔE value.

Figure 5.40 shows such a plot for a Xerox DocuColor 12 laser printer. This plot suggests that, for this printer, there is no appreciable gain in increasing the grid size beyond $s = 8$.

The extension to CMYK printers is straightforward. Note, however, that the lattice size (hence, the number of measurements) increases as s^4 and can quickly become prohibitively large. One method of improving the trade-off between lattice size and accuracy is sequential interpolation, described next.

5.10.3.2 *Sequential interpolation*

The general framework for sequential interpolation (SI) was introduced in Section 5.4.6. Here, we describe a specific application to CMYK characterization. Consider a decomposition of CMYK space into a family of CMY subspaces corresponding to different levels of K, as shown in Figure 5.41. If we were to print and measure the CMYK nodes of each CMY lattice, we would obtain a series of volumes in $L^*a^*b^*$ space, as illustrated schematically in the same figure. Each gamut volume corresponds to variations in C, M, and Y, with fixed K. Note that as K increases, the variation in color, and hence the gamut volume, decreases. For the case where K = 100%, we have almost negligible color variation. The fact that the curvature of the function strongly depends on K motivates an SI structure comprising a family of CMY

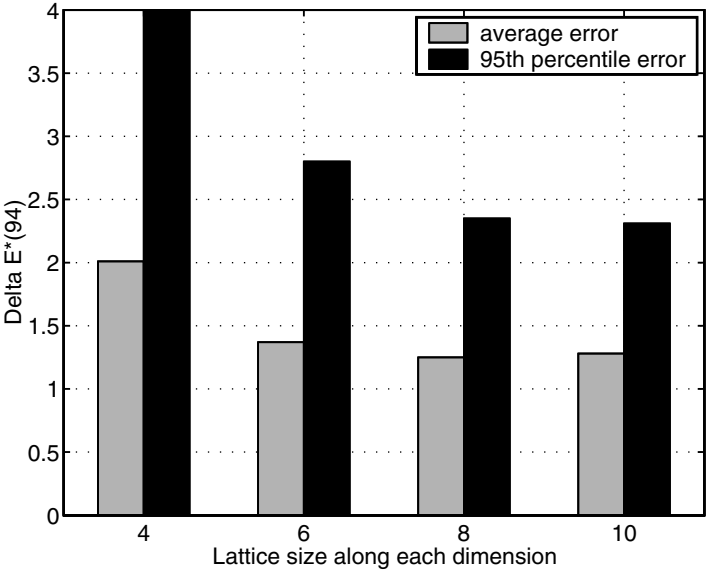


Figure 5.40 ΔE vs. lattice size.

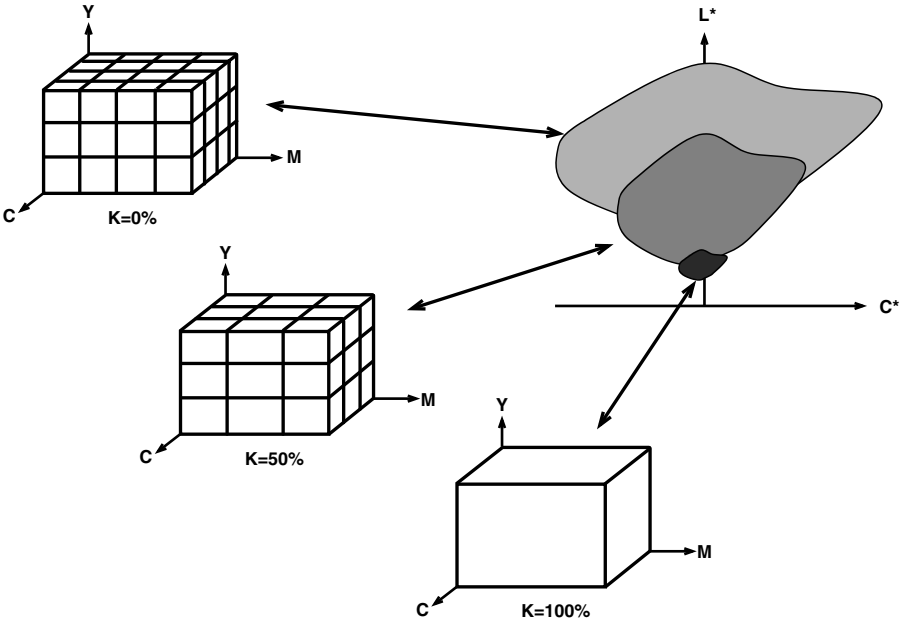


Figure 5.41 Sequential interpolation: a decomposition of CMYK into a family of CMY subspaces at different K and corresponding CIELAB gamuts. The CMY lattices become coarser as K increases.

lattices for different K. A finely sampled CMY lattice is used for K = 0, and the lattice size decreases with increasing K, as shown in Figure 5.41. When building the SI structure, each CMY lattice is filled with measured CIELAB values. Interpolation to map CMYK to CIELAB is performed as follows:

- Project the input CMYK point onto the K dimension and select neighboring levels K_j and K_{j+1}.
- Project the input CMYK point onto CMY space and perform three-dimensional interpolation on the two CMY lattices corresponding to levels K_j and K_{j+1} to produce two CIELAB points.
- Use the input K value to perform one-dimensional interpolation of these two CIELAB points.

Table 5.2 shows experimental results comparing the SI structure with a regular lattice. For approximately the same lattice size, the SI technique offers superior accuracy, hence improving the quality/cost trade-off. Further details are given by Balasubramanian.[72]

It is noteworthy that the standard IT8.7/3 printer characterization target described in Section 5.3 facilitates SI. The target contains 6 CMY lattices of size 6^3, 6^3, 5^3, 5^3, 4^3, 2^3, corresponding to K values (in percentage) of 0, 20, 40, 60, 80, 100, respectively.

Table 5.2 Comparison of Accuracy and Number of Training Samples for Standard vs. Sequential Interpolation

Model	CIE '94 ΔE		Number of LUT Nodes
	Average	95th Percentile	
Regular 4 × 4 × 4 × 4 lattice	3.0	12.3	256
Sequential interpolation with 5^3, 4^3, 3^3, 2^3 CMY lattices corresponding to k = 0, 85, 170, 255	1.8	6.25	224

5.10.3.3 *Other empirical approaches*

Tominaga (Chapter 9 of Reference 7) describes an example of a neural network for printer characterization. This is accomplished in two steps. First, a four-layer neural net is derived for the forward transform from CMYK to CIELAB using over 6500 training samples. Next, a cascaded eight-layer neural net is constructed, the first stage being the inverse mapping from CIELAB to CMYK and the second stage being the previously derived forward mapping from CMYK to CIELAB. The second stage is kept static, and the first stage is optimized to minimize the CIELAB-to-CIELAB error for the overall system. Tominaga reports an average ΔE_{ab} of 2.24 for a dye sublimation printer. As with the other techniques, the optimal number of training samples and the neural net structure depend on the printer characteristics and desired accuracy, and they have to be initially determined by trial and error.

Herzog[73] proposes an analytic model for the mapping between CMY and CIELAB. The printer gamut is described as a family of nested shells in both CMY and CIELAB space. A simple mathematical model of distortion and scaling operations is used to relate each shell from one space to another via an intermediate representation called a *kernel gamut*. Colors in between the shells are mapped via linear interpolation. A total of 626 measurements are required to derive the model, and average ΔE_{ab} errors between 0.7 and 2.5 are reported for various data sets.

5.10.4 *Hybrid approaches*

We have seen thus far that physical models and empirical techniques offer different trade-offs between effort and accuracy. There are two ways to combine the strengths of these two classes of techniques. The first is to use empirical data to optimize the parameters of a physics-based model. Many examples of this were encountered in the optimization of BB, KM, and Neugebauer models. The second is to use empirical data to refine the prediction of a printer model as a post-processing step, as shown in Figure 5.42. The assumption is that the model is a good first-order approximation, and that a small number of additional refinement samples is sufficient to correct for objectionable inaccuracies in the model.[12] The number and distribution

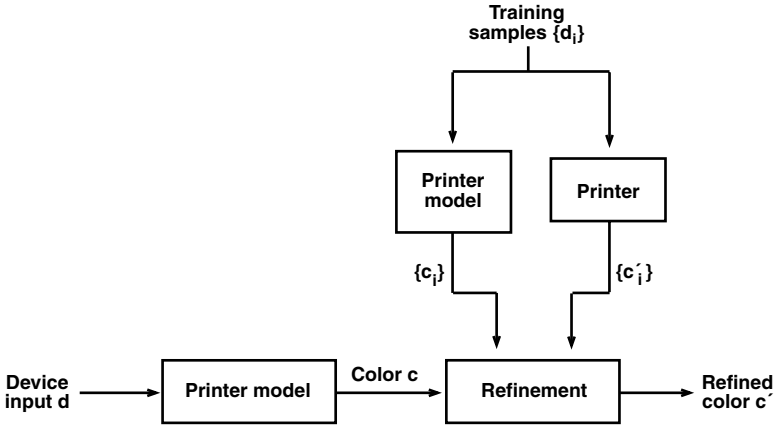


Figure 5.42 Block diagram showing refinement of printer model.

of refinement samples depend on the characteristics of the printer and the model, as well as on accuracy requirements. If the printer model is known to be erroneous in certain regions of color space, the refinement samples can be chosen with a denser sampling in these regions. Similarly, regions of color space to which the human visual system is more sensitive (e.g., flesh tones and neutral colors) can be sampled more densely. In the absence of such information, a reasonable approach is to span the gamut with an approximately uniform sampling.

In the case of forward printer characterization, the refinement is a colorimetric function from, for example, CIELAB to CIELAB. Any of the multidimensional data-fitting or interpolation techniques described in Section 5.4 can be applied to estimate this function from the refinement samples. Local linear regression has been used successfully by the author[12] to reduce average ΔE_{ab} errors from approximately 5 to 2.5.

5.10.5 *Deriving the inverse characterization function*

The inverse printer characterization is a mapping from CIE color to device colorant values that, when rendered, will produce the requested CIE color under defined viewing conditions. This mapping is usually implemented as a three-dimensional LUT, so it needs to be evaluated at nodes on a regular three-dimensional lattice in CIE coordinates. Some of the lattice nodes will lie outside the printer gamut; we assume that these points are first mapped to the gamut surface with a gamut-mapping step (described in Chapter 10). Hence, we restrict the inversion process to colors that are within the printer gamut.

In the case where the forward function is described by an analytic model, a possible approach is to directly invert the parameters of the model via analytic or search-based techniques. The most notable efforts in this direction

have been in the inversion of the Neugebauer model to estimate dot area coverages from colorimetric values.[74,75] Here, we adopt a more general inversion process that is independent of the technique for determining the forward function. The process is accomplished in two steps.

1. Use the forward characterization function to generate a distribution of training samples $\{\mathbf{c}_i, \mathbf{d}_i\}$ in device-independent and device-dependent coordinates.
2. Derive the inverse function by interpolating or fitting this data.

5.10.5.1 *CMY printers*

In the case of three-colorant devices, the forward function from CMY to colorimetric coordinates (e.g., CIELAB) is a unique mapping; hence, a unique inverse exists. Any of the interpolation or fitting techniques described in Section 5.4 can be used to determine the inverse function from the training samples. Tetrahedral inversion, described in Section 5.4.5, can be used if the device signals are generated on a regular lattice. Figure 5.43 compares four fitting algorithms (local linear regression, neural network, polynomial regression, and tetrahedral inversion) as to their ability to invert a Neugebauer model derived for a Xerox DocuColor12 laser printer. The neural network

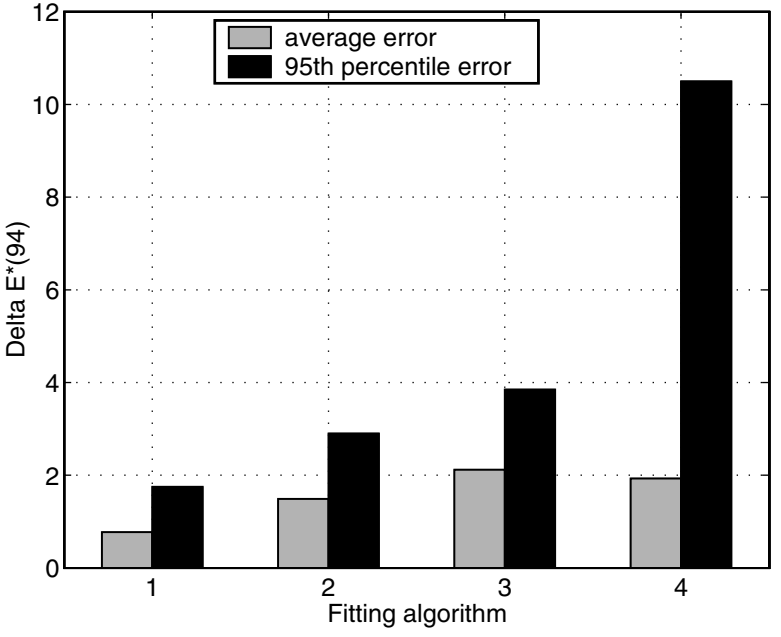


Figure 5.43 Comparison of various algorithms used to invert a Neugebauer model: 1. local linear regression, 2. neural network, 3. polynomial regression, and 4. tetrahedral inversion.

used a feed-forward algorithm with one hidden layer containing six neurons. The polynomial regression used a 3×11 matrix as in Equation 5.18b. A training set of $10^3 = 1000$ samples was used to derive the parameters for each of the fitting algorithms. An independent set of 125 samples was used as the test set. The test data, specified in CIELAB, were mapped through a given inverse algorithm to obtain CMY, which was then mapped through the forward printer model to obtain reproduced CIELAB values. The plot in Figure 5.43 shows the average and 95% ΔE^*_{94} errors between the original and reproduced values. Local linear regression and the neural network offer the best performance. In the author's experience, this observation holds generally true for a wide variety of printers. Local linear regression possesses the added advantage that it is less computationally intensive than the neural network.

Another factor that affects the overall inversion accuracy is the size of the three-dimensional LUT used to finally approximate the inverse function. An experiment was conducted to study overall inversion error as a function of LUT size. The workflow is the same as described in the preceding paragraph, except that the inverse function is now a three-dimensional LUT built using local linear regression on 1000 training samples. Figure 5.44 is a plot of overall inversion error as a function of LUT size. The error decreases with increasing LUT size; however, beyond a certain point, the returns diminish. From the plot, it is clear that a LUT size beyond $16 \times 16 \times 16$ does not afford a noticeable gain in accuracy — another observation that has been seen to

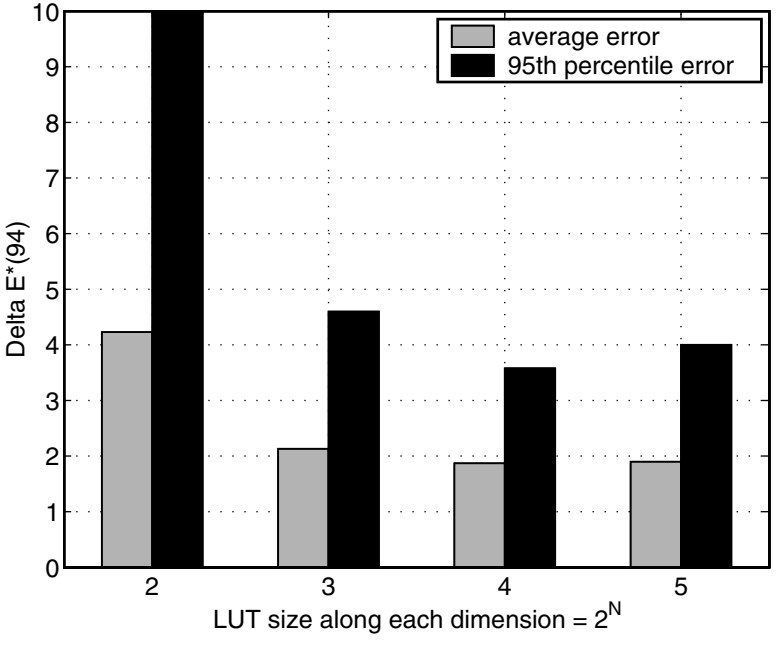


Figure 5.44 LUT approximation error vs. LUT size.

hold true for a wide variety of printers. Note that the relative spacing of nodes along each dimension can also affect LUT accuracy. In this experiment, the nodes were spaced uniformly, because the input space, CIELAB, in which the LUT was built, is approximately visually uniform.

5.10.5.2 CMYK printers

Although, in principle, the three C, M, and Y colorants suffice to produce all perceivable hues, very often, a fourth black (K) colorant is used for several reasons. First, the K colorant is usually considerably less expensive than C, M, and Y, and it can thus be used in lieu of CMY mixtures to render dark neutrals and shadows. Second, the addition of K can result in an increase in gamut in the dark regions of color space in comparison to what is achievable using only CMY mixtures. Third, the use of K can help reduce the total amount of colorant required to produce a given color, a feature that is critical in certain technologies such as inkjet printing.

In the context of device characterization, the K colorant introduces redundancy into the forward transform, as a large (in principle, infinite) number of CMYK combinations can result in the same colorimetric measurement. This results in the inverse function being ill posed, and additional constraints are required to generate a unique CMYK combination for each input CIE color. Some common methods of deriving the constrained inverse are presented next.

5.10.5.2.1 Inversion based on K addition, undercolor removal, and gray component replacement. The processes of black (K) addition, undercolor removal (UCR), and gray component replacement (GCR) trace their origins to the graphic arts printing industry.[50] Together, they define a unique transform from a set of canonical CMY primaries to the CMYK signals for the given printer. Geometrically, the transform generates a three-dimensional manifold within the four-dimensional CMYK space, with the property that every CMYK combination within the manifold results in a unique colorimetric response. Once this transform is established, the inversion can be carried out on the canonical CMY device as described in Section 5.10.5.1. Figure 5.45 shows the derivation and application of the inverse function for a CMYK printer. The two functions in Figure 5.45b are usually concatenated into a composite inverse transform from CIE to CMYK signals. Recall that the printer is assumed to have been calibrated, so the CMYK signals resulting form the inversion process are finally processed through the calibration functions prior to printing. In some implementations, the calibration is concatenated with the characterization or stored in the same profile.

There are numerous methods for designing K addition, UCR, and GCR functions. They are usually chosen for an optimal trade-off among factors such as gamut volume, colorant area coverage, and smoothness of transitions from neutral to non-neutral colors. The trade-off is usually carried out

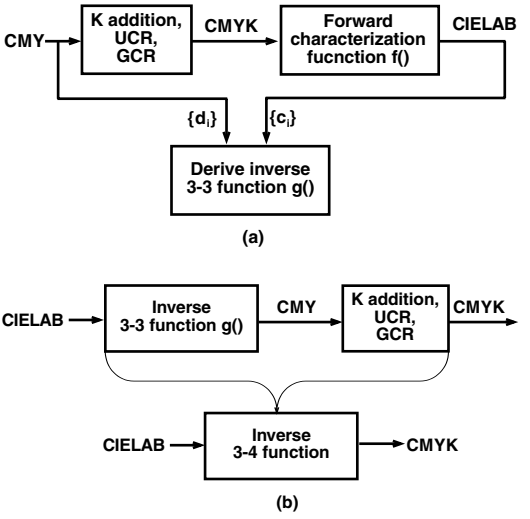


Figure 5.45 Constrained inverse characterization of CMYK printers: (a) construction of inverse 3–3 function and (b) combining the inverse 3–3 function with K addition, UCR, and GCR to construct inverse 3–4 function.

heuristically with knowledge of the printer characteristics and quality requirements. Some examples of these functions are presented next.

5.10.5.2.1.1 Black addition. This is commonly chosen to meet a desired behavior along the C = M = Y axis. Suppose the printer has been gray-balanced and linearized to neutral L^*. If we define darkness D^* as a scaled inverse of L^* using Equation 5.58, then we have C = M = Y = D^* along the neutral axis for the range $0 \le D^* \le D^*_{max}$. Here, D^*_{max} is the maximum digital count (e.g., 255 for an 8-bit system). We can then define K as a monotonic increasing function f_1 of neutral D^*. Numerous functional representations can be used, for example the power-law,

$$f_1(D^*) = \begin{cases} D^*_{max}\left(\dfrac{D^* - D^*_{offset}}{D^*_{max} - D^*_{offset}}\right)^{\gamma} & \text{if } D^*_{offset} < D^* \le D^*_{max} \\ 0 & \text{if } (0 \le D^* \le D^*_{offset}) \end{cases} \tag{5.96}$$

Here, γ and D^*_{offset} are parameters that can be adjusted to suit the desired behavior of K along the neutral axis. For γ > 1, larger values of γ and D^*_{offset} result in less aggressive f_1 (i.e., less K is used for a given amount of neutral C = M = Y). As γ and D^*_{offset} approach 1 and 0, respectively, f_1 becomes more aggressive, with the amount of K approaching the amount of neutral C = M = Y.

5.10.5.2.1.2 Undercolor removal. This function describes the amount of reduction in CMY primaries to compensate for the K addition. It is also derived with attention to the neutral axis. A simple form of CMY reduction is given by

$$C' = C - f_2(D^*) \tag{5.97}$$

with analogous expressions for M and Y. Again, we are abounded with numerous strategies for $f_2(D^*)$. One approach is based on the rationale that the CMY reduction should be proportional to the amount of K addition,

$$f_2(D^*) = \alpha f_1(D^*), \quad 0 \le \alpha \le 1 \tag{5.98}$$

The case where $\alpha = 1$ (i.e., CMY subtraction equals K addition) is often referred to as 100% UCR.

A more sophisticated approach is to derive f_2 to colorimetrically compensate for the K addition. This can be performed as follows. For a given neutral input C = M = Y sample, the resulting L^* and hence D^* that would be produced by printing this sample can be predicted via the forward characterization function. The amount of K associated with this input C = M = Y is given by $f_1()$. We can now derive the new smaller amounts, C′ = M′ = Y′, which produce the same D^* when combined with the given K. This step is achieved by combining different C = M = Y levels with the given K, running through the forward transform, and picking the combination that produces the desired D^*. Finally, f_2 is the difference between the original C = M = Y and final C′= M′= Y′.

A key factor to be considered in choosing black addition and UCR parameters is the total area coverage (TAC) that is permissible for the given device, especially in the dark portions of the gamut. For many CMYK printers, TACs near 400% will result in defects (e.g., ink bleeding in inkjet printers or improper toner fusing and flaking in xerographic printers). Hence, an appropriate limit must be placed on TAC, and this in turn affects the K addition and UCR parameters. If the colorimetric approach described in the preceding paragraph is adopted, accuracy will likely be sacrificed toward the dark end of the gamut due to TAC limits.

5.10.5.2.1.3 Gray component replacement. Thus far, K addition and UCR have been defined for neutral samples C = M = Y. GCR is a generalization of these functions for the entire gamut of CMY combinations. A fundamental assumption is that the gray component of an arbitrary CMY combination is given by the minimum of C, M, Y. This gray component can then be used as input to the K addition and UCR functions.

$$X = \min(C,M,Y)$$

$$K = f_1(X)$$

$$C' = C - f_2(X)$$

$$M' = M - f_2(X)$$

$$Y' = Y - f_2(X) \tag{5.99}$$

Clearly, one can conceive numerous enhancements to this simple model. The CMY subtraction can be performed in other spaces such as optical density. This can be accomplished in the current framework by applying a transform to the chosen space before CMY subtraction and applying the inverse transform after subtraction. Second, functions f_1 and f_2 can be multidimensional functions that depend on more than just the minimum of C, M, Y. This may be desirable if, for example, the optimal balance between K and CMY along the neutral axis is different from that along the edges of the gamut. Finally, CMY reduction can be accomplished by methods other than simple subtraction. For another perspective on UCR and GCR techniques, refer to Holub et al.[76]

5.10.5.2.2 Direct constraint-based CMYK inversion. The previous approach used a CMY-to-CMYK transform to arrive at a constrained inverse. A more general and direct method is to obtain the set of all CMYK combinations that result in the given input CIE color and select a combination that satisfies certain constraints. Examples of such constraints include:

1. Total colorant area coverage (i.e., C + M + Y + K) is less than a threshold.
2. The amount of K with respect to the minimum and maximum K that can produce the given color is constrained.
3. Stability is maximized (i.e., change in colorant values results in minimum change in CIE color).
4. Smoothness is maintained with respect to neighboring colors in CIE space.
5. Gamut volume is maximized.
6. Spatial artifacts such as misregistration and moiré are minimized or constrained.

Constraint 5 implies that, if a color is achievable with only one CMYK combination, this should be used, even if some of the other constraints are not met. Space considerations do not permit us to elaborate on the other constraints. We refer the reader to Mahy[77] for detailed discussions of constraints 1 through 4), and Balasubramanian et al.[78] for a description of how UCR and GCR are optimized to minimize moiré. Cholewo[79] describes another constrained inversion technique that takes gamut mapping into account.

Note that the aforementioned two approaches can be combined. For example, the UCR/GCR approach can be used to generate an initial CMYK combination, which can then be refined iteratively to meet one or more of constraints 1 through 6.

5.10.6 Scanner-based printer characterization

All the foregoing discussion implicitly assumes the use of a colorimeter or spectrophotometer for color measurement. Another approach is to use a color scanner as the measurement device in printer characterization. The main advantage of this approach is that scanning a color target is less labor intensive than spectrophotometric measurement; furthermore, the measurement time does not depend on the number of patches on the target. Because a scanner is generally not colorimetric, it must first be characterized. Fortunately, the characterization needs to be derived only for the colorant-medium combination used by the given printer, hence empirical techniques can be used with very accurate results (see Section 5.6.3). The scanner characterization produces a transform from scanner RGB to colorimetric or spectral coordinates, thus turning the scanner into a colorimetric device for the given colorants and medium. Printer characterization can then be carried out as described in the foregoing sections, with the target measurement step being replaced by scanning of the target followed by the necessary image processing (i.e., mapping the scanned image through the scanner characterization and extracting colorimetric values of each patch). Note that this approach intimately links the scanner and printer into a characterized pair.

5.10.7 Hi-fidelity color printing

The term *high-fidelity (hi-fi) color* refers to the use of extra colorants in addition to the standard C, M, Y, K. Two strategies can be adopted for hi-fi color printing. In one approach, the additional colorants are of the same hues as the standard colorants but of different concentrations. Usually, C and M are chosen for this. The use of multiple concentrations allows for superior rendering of detail in the highlights and shadows as well as smoother transitions from highlights to mid-tones to shadows. In the second strategy, the extra colorants are of hues that are different from C, M, Y. One purpose of this is to extend the color gamut. Due to unwanted absorptions, printer gamuts are often deficient in saturation and lightness in the secondary overprints, i.e., R, G, B. Hence, hi-fi colorants are chosen to extend the gamut in these regions of color space, as shown in the schematic in Figure 5.46. Another purpose is to reduce metamerism and enable spectral reproduction by using colorants with relatively narrowband spectral characteristics (see Section 5.11).

5.10.7.1 Forward characterization

For forward characterization, the BB, KM, and Neugebauer models all extend in a straightforward manner to an arbitrary number of colorants. In the case of BB and KM, the number of measurements increases linearly with the number of colorants while, for the Neugebauer model, the number of measurements increases exponentially due to the need for including solid overprints. In the latter case, the number of overprints can become prohibitively large; hence, a two-stage model can be employed. The BB or KM is used to predict

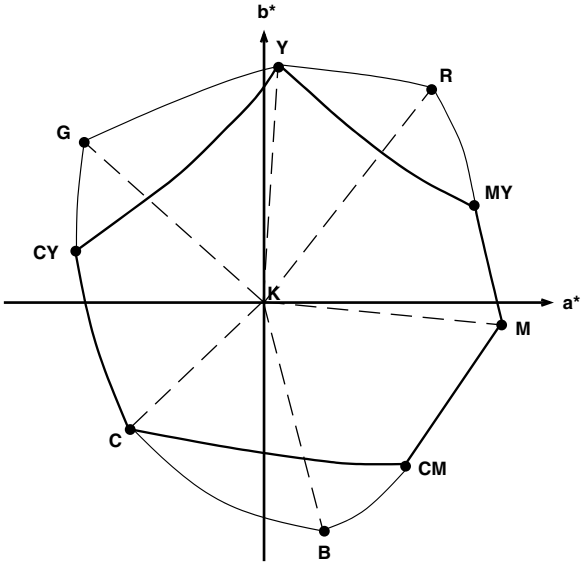


Figure 5.46 Use of R, G, B hi-fi colorants to extend the gamut achieved with standard CMYK printing.

the reflectances of solid overprints, and this is fed to the Neugebauer model, which predicts the reflectances of arbitrary colorant mixtures. The reader is referred to Section 6.6 of Reference 7 for further discussions and references.

Recently, a novel hi-fi color modeling approach has been proposed by Van de Capelle and Meireson (Reference 7, Chapter 10) that obviates the need for measuring overprints even for halftone processes. The main advantage is therefore the substantial savings in number of measurements as the number of colorants increases. Single-colorant stepwedges between 0 and 100% area coverage are printed under three conditions:

1. On the naked substrate
2. On the substrate with 50% black
3. On the substrate with 100% black

From these measurements, three substrate-independent parameters are estimated for each colorant, namely scattering, interaction, and absorption. These parameters describe spectral properties of the colorants independent of the underlying substrate and can be used to predict the reflectance of a colorant layer on any given substrate. The reflectance of an n-layer mixture of colorants is modeled iteratively by calculating the reflectance of the $n - 1$ layer mixture and treating this as the substrate for the nth layer. The authors report average ΔE_{ab} between 1.8 and 3.0 for various data sets. See the aforementioned reference for details.

5.10.7.2 *Inverse characterization*

As with CMYK printing, hi-fi color introduces redundancy into the color reproduction process in that many colorant combinations can result in the same perceived (CIE) color. The inversion process must select a unique colorant combination for each input CIE color. Additional considerations include minimization of colorant area coverage, minimization of moiré in the case of rotated halftone screening, and a smooth characterization function from CIE to device signals.

A common strategy is to partition the color gamut into subgamuts formed from combinations of three or four colorants. Referring to Figure 5.46, one approach is to partition the gamut into six subgamuts formed from the following colorant combinations: CGK, GYK, YRK, RMK, MBK, and BCK. Here, R, G, and B represent hi-fi colorants chosen in the red, green, and blue hues.[80,81] (Often, orange is used instead of red; the idea still holds.) Because C + Y combinations are spectrally redundant with G, the former are disallowed, and only C + G and G + Y combinations are entertained. This rule guarantees uniqueness, and is applied likewise for the other combinations. Also, because only three screens angles are being used to render any given input color, moiré can be handled using the same rules applied for conventional printing. Another variant[82] represents the gamut as overlapping subgamuts formed by GYRK, YRMK, RMBK, MBCK, BCGK, and CGYK. Several criteria, including minimization of colorant area coverage and smoothness of transitions across subgamuts, are used to select unique colorant combinations for a given input color. A third variant[83] employs an approach directly akin to UCR and GCR for CMYK printing. The gamut is partitioned into the subgamuts formed by YRMK, MBCK, and CGYK. In the YRMK subgamut, the M and Y signals are fed to an R addition function and MY subtraction function. Analogous operations are applied in the other subgamuts. The functions are chosen to produce a unique colorant combination for each input color while maximizing the volume of each subgamut. As with conventional UCR/GCR, the latter can be aided by applying a nonlinearity to the input signal and inverting this after the addition and subtraction operations have been performed. Note that, with all these approaches, hi-fi colorants are used to render colors that can also be achieved with standard CMYK colorants. In a fourth variant, only CMYK colorants are used to achieve colors within the standard CMYK gamut, and hi-fi colorants are introduced only in those regions of color space that cannot be reproduced with CMYK mixtures. This approach offers better compatibility with standard CMYK devices. Details of these techniques are deferred to the stated references.

5.10.8 *Projection transparency printing*

Overhead transparency projection continues to be a common medium for communication of color information. Characterization for this application is complicated by several factors. First, the final viewing conditions and

projector characteristics are difficult to predict *a priori* and may be very different from those used for characterization. Second, it is difficult to achieve strict spatial uniformity, especially when the image is projected into a large screen. Fortunately, color accuracy requirements for projected transparency are usually not so stringent as to require careful characterization. But, if accuracy is important, care must be taken to control the aforementioned conditions as well as possible.

The characterization procedure is conceptually the same as for conventional printers. A test target is printed on a transparency and projected under representative viewing conditions. The main factor that affects the latter is ambient lighting. The projected image is measured with a spectroradiometer. To minimize light scattering, the transparency can be masked to present only one patch at a time to the spectroradiometer. The measured CIELAB values and original printer coordinates are then used to build the characterization transform. For all the reasons stated above, a simple transparency model such as the Beer–Bouguer formula in Equation 5.64 may not suffice to predict the characterization; rather, empirical approaches and three-dimensional LUTs may be necessary to capture the various complex effects. The techniques described in Section 5.4 and Section 5.10.3 can be used for this purpose. For further reading, see Chapter 10 of Kang[48] and the work by Cui et al.[84]

5.11 Characterization for multispectral imaging

The discussion in this chapter is primarily based on colorimetric reproduction wherein the device-independent representation of Figure 5.1 comprises three color channels. An emerging area of research that is gaining increasing attention is multispectral imaging, whereby the goal is to capture, store, and reproduce narrowband spectral information rather than three-dimensional colorimetric data. A primary motivation for preserving spectral information is that it mitigates the metamerism problems encountered in conventional colorimetric reproduction. A related advantage is that colorimetric characterizations can be computed dynamically for different viewing illuminants from one set of spectral data. A schematic of a spectral imaging system is shown in Figure 5.47. The input device records color in N channels, where N is greater than three. This is usually accomplished by using N narrowband filters in the image acquisition device. Spectral information is reconstructed from the data via spectral characterization of the input device.

A major concern in spectral imaging is the substantial increase in the amount of data to be handled (i.e., from 3 to 30 or more channels). This necessitates an efficient encoding scheme for spectral data. Most encoding techniques are based on the well-known fact that spectra found in nature are generally smooth and can be well approximated by a small number (i.e., between five and eight) of basis functions.[85] The latter can be derived via principal-component analysis (PCA) described in an Chapter 1. PCA yields a compact encoding for spectral data and can serve as the device-independent color space in a multispectral imaging framework. An important con-

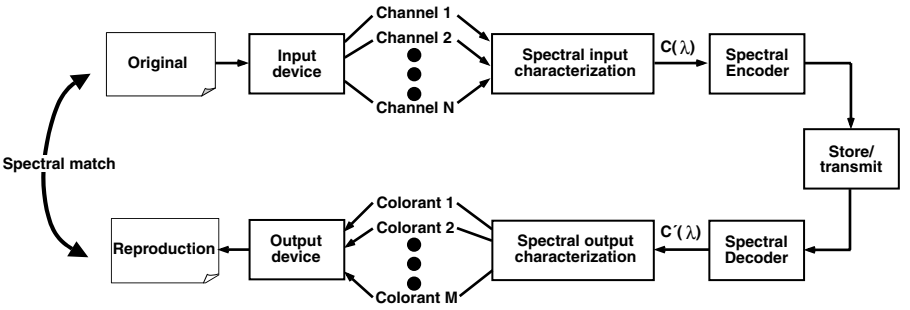


Figure 5.47 Multispectral imaging system.

sideration in selecting the PCA encoding is to ensure compatibility with current colorimetric models for color management.[86]

The goal of the output device is to reproduce the spectral (rather than colorimetric) description of the input image via a spectral characterization. As with input devices, a "spectral printer" must employ more than the traditional C, M, Y colorants to facilitate spectral reproduction. The forward characterization transform can be achieved using many of the techniques described in this chapter; derivation of the inverse transform is, however, a more challenging problem.

For further details, the reader is referred to the book by MacDonald and Luo[7] for a description of multispectral image capture and encoding techniques; the work by Tzeng and Berns[87–89] for contributions to multispectral printing, including colorant selection and characterization algorithms; and Rosen et al.[90] for a description of a framework for spectral characterization.

5.12 *Device emulation and proofing*

Frequently, it is desired to emulate the color characteristics of one device on another. Two classic examples are the use of a proof printer to emulate a color press and the use of a softcopy display to emulate a printer. (The latter is known as *softproofing*.) In both cases, the idea is to use a relatively inexpensive and easily accessed device to simulate the output of a device that is less accessible and for which image rendering is costly. We will generically refer to these two devices as the proofing device and target device, respectively. Device emulation is a cost-effective strategy when iterative color adjustments on an image are needed prior to final production. The iterations are carried out on the proofing device, and the final production takes place on the target device.

Figure 5.48 is a flow diagram showing how a device emulation transform can be generated from characterizations of the proofing and target devices. A device-independent color input $\mathbf{c}$ is transformed to the target device space $\mathbf{d}_t$ via the function $g_t()$. This transform should be the same as the one used for final rendering of images on the target device. Because the target device

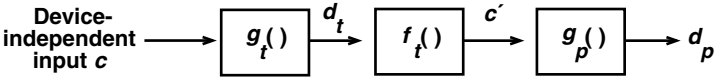


Figure 5.48 Block diagram for device emulation. Functions $f()$ and $g()$ denote forward and inverse characterizations. Subscripts "t" and "p" refer to target and proofing devices. Device-independent and device-dependent color representations are denoted $\boldsymbol{c}'$ and $\boldsymbol{d}$.

is invariably a printer, $\mathbf{d}_t$ is usually a CMYK representation. Next, the device colors are transformed back to device-independent coordinates $\mathbf{c}'$ via the target device's forward transform $f_t()$. Thus, $\mathbf{c}'$ describes the appearance of the given input color on the target device. The remaining step is to match this color on the proofing device. This is accomplished by applying the inverse transform $g_p()$ for the proofing device and rendering the resulting device coordinates $\mathbf{d}_p$ to this device. Depending on whether the proofing device is a printer or display, $\mathbf{d}_p$ will be in CMYK or RGB space, respectively. For efficiency, the operations in Figure 5.48 are usually concatenated into a single emulation transformation.

In an alternative scenario, the input image may already exist in the target device space $\mathbf{d}_t$, in which case only the last two blocks in Figure 5.48 need to be executed. A common example of this occurs when CMYK files prepared for a standard offset press (e.g., SWOP) are to be rendered on a digital CMYK proofing device. The result is a four-dimensional CMYK-to-CMYK transform.

The four-to-four transform deserves brief mention. We learned in Section 5.10.5.2 that constraints are needed to produce a unique output CMYK combination for each distinct input color, and we introduced the notion of K addition, UCR, and GCR for this purpose. In the case where the input is a CMYK space, an alternative constraint might be to determine the output K as a function of input K. The simplest instantiation is to simply preserve the input K signal. Techniques of this type are presented by Cholewo[91] and Zeng.[92]

5.13 *Commercial packages*

A number of calibration and characterization products are available that offer a wide range of capabilities and performance. Comprehensive product descriptions are beyond the scope of this chapter. However, for the more application-oriented reader, the following is a list of mainstream color management products available at the time this book was published.

Note that this list mainly includes stand-alone packages and that calibration and characterization functions are also often embedded within device controllers (e.g., print controller products by Creo-Scitex, Electronics for Imaging) or as applications bundled with current operating systems (e.g., Adobe Gamma for display calibration).

GretagMacbeth (ProfileMaker)	www.gretagmacbeth.com
Agfa (ColorTune)	www.agfa.com/software/colortune.html
ColorBlind (Matchbox)	www.color.com
ColorSavvy (WiziWYG Pro)	www.colorsavvy.com
Kodak (ColorFlow)	www.kodak.com
LinoColor (Scanopen, Viewopen, Printopen)	www.linocolor.com
Monaco Systems (MonacoEZcolor and MonacoProfiler)	www.monacosystems.com
Praxisoft (WiziWYG, CompassProfile)	www.praxisoft.com/products/cms.html

5.14 Conclusions

In this chapter, we hope to have provided the reader with the theoretical foundation as well as practical procedures and guidelines to accomplish device calibration and characterization. The chapter began with a general conceptual overview and terminology associated with color characterization of input and output devices. Two basic approaches to characterization were presented: model-based and empirical. In addition, hybrid techniques were described that combine the strengths of both approaches. Next, a treatment was given of fundamental topics that apply to all forms of device characterization — namely color measurement technology, data interpolation and fitting algorithms, and quantitative analysis tools. This was followed by a detailed discussion of the calibration and characterization of several common input and output devices, including scanners, digital cameras, displays, and printers. Finally, the chapter concluded with several special topics, namely characterization of hi-fi and projection transparency printers, device emulation techniques, and commercial color characterization products.

Clearly, there are many aspects to this subject, and we have not been able to cover all of them in great depth. It is hoped that the extensive set of references will serve for further enquiry into any given topic. Finally, it must be emphasized that device characterization is not a topic that stands on its own, and it cannot by itself guarantee high-quality results in a color imaging system. The latter calls for a thorough system-wide understanding of all the components in the color imaging chain and their interactions. This is evidenced by the numerous cross references to other chapters in this book.

Acknowledgment

The author wishes to thank Dean Harrington for his assistance in preparing the figures for this chapter.

References

1. *Postscript Language Reference Manual*, 2nd ed., Addison-Wesley, Reading, MA, Chap. 6.

2. Trussell, H. J., DSP solutions run the gamut for color systems, *IEEE Signal Processing*, 10, 8–23, 1993.
3. Luther, R., Aus Dem Gebiet der Farbreizmetrik, *Z. Tech. Phys.*, 8, 540–558, 1927.
4. Ives, H. E., The transformation of color-mixture equations from one system to another, *J. Franklin Inst.*, 16, 673–701, 1915.
5. Sharma, G. and Trussell, H. J., Color scanner performance trade-offs, in *Proc. SPIE, Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts*, J. Bares, Ed., Vol. 2658, 1996, 270–278.
6. Sharma, G. Trussell, G, H. J, and Vrhel, M. J., Optimal non-negative color scanning filters, *IEEE Trans. Image Proc.*, 7(1), 129–133, 1998.
7. MacDonald, L. W. and Luo, M. R., Eds., *Colour Imaging — Vision and Technology,* Wiley, Chichester, U.K., 1999.
8. Alessi, P. J. and Cottone, P. L., Color reproduction scheme for Kodak Organic Light Emitting Diode (OLED) technology, *AIC Color 01*, Rochester, NY, June 24–29, 2001.
9. Rich, D., Critical parameters in the measurement of the color of nonimpact printing, *J. Electronic Imaging*, 2(3), 1993, 23–236, 1993.
10. Zwinkels, J. C., Colour-measuring instruments and their calibration, *Displays,* 16(4), 163–171, 1996.
11. Rolleston, R., Using Shepard's interpolation to build color transformation tables, in *Proc. IS&T/SID's 2nd Color Imaging Conference*, November 1994, 74–77.
12. R. Balasubramanian, Refinement of printer transformations using weighted regression, in *Proc. SPIE, Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts*, J. Bares, Ed., Vol. 2658, 1996, 334–340.
13. Hung, P-C., Colorimetric calibration in electronic imaging devices using a look-up table model and interpolations, *J. Electronic Imaging*, 2(1), 53–61, 1993.
14. Chang, J. Z., Allebach, J. P., and Bouman, C. A., Sequential linear interpolation of multidimensional functions, *IEEE Trans. on Image Processing*, Vol. IP-6, September 1997, 1231–1245.
15. T. Masters, *Practical Neural Network Recipes in C++,* Academic Press, San Diego, CA, 1993.
16. Farin, G., *Curves and Surfaces for Computer Aided Geometric Design — A Practical Guide*, Academic Press, San Diego, CA, 1988.
17. Press, H., Flannery, B. P., and Vetterling, W. T., *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K., 1988.
18. Luo, M. R., Cui, G. and Rigg, B., The development of the CIEDE2000 colour-difference formula, *Color Res. Appl.*, 25, 340–350, 2001.
19. Engeldrum, P. G., *Psychometric Scaling*, Imcotek Press, Winchester, MA, 2000.
20. Sharma, G. and Trussell, H. J., Figures of merit for color scanners, *IEEE Trans. Image Proc.*, 6(7), 990–1001, 1997.
21. Quan, S. and Ohta, N., Optimization of camera spectral sensitivities, in *Proc. IS&T/SID's 8th Color Imaging Conference*, November 2000, 273–278.
22. Knox, K. T., Integrating cavity effect in scanners, in *Proc. IS&T/OSA Optics & Imaging in the Information Age*, Rochester NY, 1996, 156–158.
23. Kang, H. R., Color scanner calibration, *J. Imaging Sci. Technol.*, 36(2), 162–170, 1992.
24. Sharma, G. and Trussell, H. J., Set theoretic estimation in color scanner characterization, *J. Electronic Imaging*, 5(4), 479–489, 1996.
25. Wandell, B. A., *Foundations of Vision*, Sinauer Associates, Sunderland, MA, 1995.

26. Finlayson, G. D., Hordley, S., and Hubel, P. M., Recovering device sensitivities with quadratic programming, in *Proc. IS&T/SID's 6th Color Imaging Conference*, November 1998, 90–95.
27. Hubel, P. M., Sherman, D., and Farrell, J. E., A comparison of methods of sensor spectral sensitivity estimation, in *Proc. IS&T/SID's 2nd Color Imaging Conference*, November 1994, 45–48.
28. Finlayson, G. D. and Drew, M. S., The maximum ignorance assumption with positivity, in *Proc. IS&T/SID's 4th Color Imaging Conference*, November 1996, 202–205.
29. Berns, R. and Shyu, M. J., Colorimetric characterization of a desktop drum scanner using a spectral model, *J. Electronic Imaging*, 4(4), 360–372, 1995.
30. Sharma, G., Target-less scanner color calibration, *J. Imaging Sci. Technol.*, 44(4), 301–307, 2000.
31. Kang, H. R. and Anderson, P. G., Neural network applications to the color scanner and printer calibrations, *J. Electronic Imaging*, 1(2), 125–135, 1992.
32. ISO 17321 (WD4), *Graphic Technology and Photography — Colour Characterisation of Digital Still Cameras (DSCs) Using Colour Targets and Spectral Illumination*.
33. ISO 14525 (FDIS), *Photography — Electronic Still Picture Cameras — Methods of Measuring Opto-Electronic Conversion Functions (OECFs).*
34. Finlayson, G. D. and Drew, M. S., White point preserving color correction, in *Proc. IS&T/SID's 5th Color Imaging Conference*, November 1997, 258–61.
35. Hubel, P. M. et al., Matrix calculations for digital photography, in *Proc. IS&T/SID's 5th Color Imaging Conference*, November 1997, 105–111.
36. Hong, G., Han, B., and Luo, M. R., Colorimetric characterisation for low-end digital camera, in *Proc. 4th International Conference on Imaging Science and Hardcopy*, 2001, 21–24.
37. Hong, G., Luo, M. R., and Rhodes, P. A., A study of digital camera colorimetric characterisation based on polynomial modelling, *Color Res. Appl.*, 26(1), 76–84, 2001.
38. Finlayson, G. D., Drew, M. S., and Funt, B. V., Spectral sharpening: Sensor transformations for improved colour constancy, *J. Opt. Soc. Am.*, 5, 1553–1563, 1994.
39. Berns, R., S. Motta, R. J., and Gorzynski, M. E., CRT Colorimetry, Part 1: Theory and practice, *Color Res. Appli.*, 18(5), 299–314, 1993.
40. Edgar, A. D. and Kasson, J. M., Display Calibration, U.S. Patent No. 5,298,993, issued March 29, 1994.
41. Engeldrum, P. and Hilliard, W., Interactive Method and System for Color Characterization and Calibration of Display Device, U.S. Patent No. 5,638,117, issued June 10, 1997.
42. Ohara, K. et al., Apparatus for Determining a Black Point on a Display Unit and Method of Performing the Same, U.S. Patent No. 6,084,564, issued July 4, 2000.
43. Balasubramanian, R., Braun, K., Buckley, R., and Rolleston, R., Color documents in the Internet era, *The Industrial Physicist*, 16–20, 2001.
44. Gentile, R. S. and Danciu, I. M., Method to estimate the white point on a display device, U.S. Patent No. 6,023,264, issued February 8, 2000.
45. Marcu, G. et al., Color characterization issues for TFTLCD displays, in *Proc. SPIE, Color Imaging: Device-Independent Color, Color Hardcopy, and Applications VII*, Eschbach, R. and Marcu, G., Eds., Vol. 4663, 2002, 187–198.

46. Sharma, G., Comparative evaluation of color characterization and gamut of LCDs versus CRTs, in *Proc. SPIE, Color Imaging: Device-Independent Color, Color Hardcopy, and Applications VII*, Eschbach, R. and Marcu, G., Eds., Vol. 4663, 2002, 177–186.
47. Kwak, Y. and MacDonald, L. Accurate prediction of colours on liquid crystal displays, in *Proc. IS&T/SID's 9th Color Imaging Conference*, November 2001, 355–359.
48. H. R. Kang, *Color Technology for Electronic Imaging Devices*, SPIE, Bellingham, WA, 1997.
49. Bala, R., What is the chrominance of gray? *Proc. IS&T and SID's 9th Color Imaging Conference*, November 2001, 102–107.
50. Yule, J. A. C., *Principles of Color Reproduction: Applied to Photomechanical Reproduction, Color Photography, and the Ink, Paper, and Other Related Industries*, John Wiley & Sons, New York, 1967.
51. Allen, E., *Optical Radiation Measurements*, Vol. 2., *Color Formulation and Shading,* Academic Press, San Diego, CA, 1980, Chap. 7.
52. Berns, R. S., Spectral modeling of a dye diffusion thermal transfer printer, *J. Electronic Imaging*, 2(4), 359–370, 1993.
53. Parton, K. H. and Berns, R. S., Color modeling ink-jet ink on paper using Kubelka–Munk theory, in *Proc. 7th Int. Congress on Advances in Non-Impact Printing Technologies*, 1992, 271–280.
54. Kang, H. R., Kubelka–Munk modeling of ink jet ink mixing, *J. Imaging Technol.,* 17, 76–83, 1991.
55. Saunderson, J. L., Calculation of the color of pigmented plastics, *J. Optical Soc. Am.*, 32, 727–736, 1942.
56. Neugebauer, H. E. J., Die Theoretischen Grandlagen des Mehrfarben-edruckes, *Zeitschrift Wissenschaften Photography,* 73–89, 1937.
57. Rolleston, R. and Balasubramanian, R., Accuracy of various types of Neugebauer models, in *Proc. IS&T and SID's 1st Color Imaging Conference: Transforms and Transportability of Color*, November 1993, 32–37.
58. Viggiano, J. A. S., Modeling the color of multi-colored halftones, in *Proc. TAGA,* 44–62, 1990.
59. Balasubramanian, R., Optimization of the spectral Neugebauer model for printer characterization, *J. Electronic Imaging*, 8(2), 156–166, 1999.
60. Yule, J. A. C. and Nielsen, W. J., The penetration of light into paper and its effect on halftone reproduction, in *Proc. TAGA*, 65–76, 1951.
61. Maltz, M., Light scattering in xerographic images, *J. Appl. Photogr. Eng.*, 9(3), 83–89, 1983.
62. Gustavson, S. and Kruse, B., 3D modelling of light diffusion in paper, in *Proc. TAGA*, 2, 848–855, 1995.
63. Heuberger, K. J., Jing, Z. M., and Persiev, S., Color transformations and lookup tables, in *Proc. TAGA/ISCC*, 2, 863–881, 1992.
64. Arney, J. S., Engeldrum, P. G., and Zeng, H., An expanded Murray–Davies model of tone reproduction in halftone imaging, *J. Imaging Sci. Technol.*, 39(6), 502–508, 1995.
65. Lee, B. et al., Estimation of the Neugebauer model of a halftone printer and its application, in *Proc. IS&T/OSA Annu. Conf., Optics & Imaging in the Information Age,* 1997, 610–613.

66. Iino, K. and Berns, R. S., A spectral based model of color printing that compensates for optical interactions of multiple inks, AIC Color 97, in *Proc. 8th Congress International Colour Association*, 1997, 610–613.
67. Hau, C. and Huang, K., Advanced cellular YNSN printer model, in *Proc. IS&T/SID's 5th Color Imaging Conf.,* November 1997, 231–234.
68. Iino, K. and Berns, R. S., Building color management modules using linear optimization, I. Desktop color system, *J. Imaging Sci. Tech*. 42(1), 79–94, 1998.
69. Iino, K. and Berns, R. S., Building color management modules using linear optimization, II. Prepress system for offset printing, *J. Imaging Sci. Technol*., 42(2), 99–114, 1998.
70. Agar, A. U. and Allebach, J. P., An iterative cellular YNSN method for color printer characterization, in *Proc. IS&T/SID's 6th Color Imaging Conference*, November 1998, 197–200.
71. Xia, M. et al., End-to-end color printer calibration by total least squares regression, *IEEE Trans. Image Proc*., 8(5), 700–716, 1999.
72. Balasubramanian, R., Reducing the cost of lookup table based color transformations, *J. Imaging Sci. Technol*., 44(4), 321–327, 2000.
73. Herzog, P., A new approach to printer calibration based on nested gamut shells, in *Proc. IS&T/SID's 5th Color Imaging Conference*, November 1997, 245–249.
74. Pobboravsky, I. and Pearson, M., Computation of dot areas required to match a colorimetrically specified color using the modified Neugebauer equations, in *Proc. TAGA*, 65–77, 1972.
75. Mahy, M. and Delabastita, P., Inversion of the Neugebauer equations, *Color Res. Appl.,* 21(6), 404–411, 1996.
76. Holub, R., Pearson, C., and Kearsley, W., The black printer, *J. Imaging Technol.,* 15(4), 149–158, 1989.
77. Mahy, M., Color Separation Method and Apparatus for Same, U.S. Patent No. 5,878,195, issued March 2, 1999.
78. Balasubramanian, R. and Eschbach, R., Reducing multi-separation color moire via a variable undercolor removal and gray-component replacement strategy, *J. Imaging Sci. Technol*., 45(2), 152–160, 2001.
79. Cholewo, T. J., Printer model inversion by constrained optimization, in *Proc. SPIE, Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts V*, R. Eschbach and G. Marcu, Eds., Vol. 3963, San Jose, CA, 2000, 349–357.
80. Kueppers, H., Printing Process where Each Incremental Area is Divided into a Chromatic Area and an Achromatic Area and Wherein the Achromatic Areas Are Printed in Black and White and the Chromatic Areas are Printed in Color Sub-sections, U.S. Patent No. 4,812,899, issued March 14, 1989.
81. Ostromoukhov V., Chromaticity gamut enhancement by heptatone multicolor printing, in *Proc. SPIE, Device Independent Color Imaging and Imaging Systems Integration*, Motta R. J. and Berberian, H. A., Eds., Vol. 1909, 1993, 139–151.
82. Boll, H., A color to colorant transformation for a seven ink process, in *Proc. SPIE, Device Independent Color Imaging*, Walowit E., Ed., Vol. 2170, 1994, 108–118.
83. Balasubramanian, R., System for Printing Color Images with Extra Colorants in Addition to Primary Colorants, U.S. Patent No. 5,870,530, issued Feb. 9, 1999.

84. Cui, C. and Weed, S., Measurement problems for overhead projection transparency printing color calibration, in *Proc. IS&T/SID's 9th Color Imaging Conference*, November 2001, 303–309.
85. Vrhel, M. J., Gershon, R., and Iwan, L. S., Measurement and analysis of object reflectance spectra, *Color Res. Appl.*, 19(1), 4–9, 1991.
86. Keusen, T., Multispectral color system with an encoding format compatible with the conventional tristimulus model, *J. Imaging Sci. Technol.*, 40(6), 510–515, 1996.
87. Tzeng, D. Y. and Berns, R. S., Spectral-based six-color separation minimizing metamerism, in *Proc. IS&T/SID's 8th Color Imaging Conference*, November 2000, 342–347.
88. Tzeng, D. Y. and Berns, R. S., Spectral-based ink selection for multiple-ink printing I. Colorants estimation of original objects, in *Proc. IS&T/SID's 6th Color Imaging Conference*, November 1998, 106–111.
89. Tzeng, D. Y. and Berns, R. S., Spectral-based ink selection for multiple-ink printing II. Optimal ink selection, in *Proc. IS&T/SID's 7th Color Imaging Conference*, November 1999, 182–187.
90. Rosen, M. R. et al., Color management within a spectral image visualization tool, in *Proc. IS&T/SID's 8th Color Imaging Conference*, November 2000, 75–80.
91. Cholewo, T. J., Conversion between CMYK spaces preserving black separation, in *Proc. IS&T/SID's 8th Color Imaging Conference*, November 2000, 257–261.
92. Zeng, H., CMYK transformation with black preservation in color management system, in *Proc. SPIE, Color Imaging: Device-Independent Color, Color Hardcopy, and Applications VII*, Eschbach, R. and Marcu, G., Eds., Vol. 4663, 2002, 143–149.
93. Noble, B. and Daniel, J. W., *Applied Linear Algebra*, 2nd ed., Chapter 9, Prentice-Hall, Englewood Cliffs, NJ, 1977, 323–330.

appendix 5.A

Least-squares optimization

Given a $T \times m$ matrix $\mathbf{D}$ of m-dimensional input data points, and a $T \times 1$ vector $\mathbf{c}$ of one-dimensional output data points, we wish to find the optimal $m \times 1$ coefficient vector $\mathbf{a}$ that minimizes the squared error

$$E = \|\mathbf{c} - \mathbf{D}\,\mathbf{a}\|^2 \tag{5.A.1}$$

where $\|\ \|^2$ denotes the L^2 norm or vector length. We can write the error in matrix-vector notation as

$$E = [\mathbf{c} - \mathbf{D}\,\mathbf{a}]^t[\mathbf{c} - \mathbf{D}\,\mathbf{a}] = \mathbf{c}^t\mathbf{c} - 2\mathbf{c}^t\mathbf{D}\mathbf{a} + \mathbf{a}^t\mathbf{D}^t\mathbf{D}\mathbf{a} \tag{5.A.2}$$

The $\mathbf{a}$ that minimizes E is found by differentiating with respect to $\mathbf{a}$ and setting to 0.

$$\frac{\partial E}{\partial \mathbf{a}} = 2\boldsymbol{D}^t\boldsymbol{D}\boldsymbol{a} - 2\boldsymbol{D}^t\boldsymbol{c} = 0 \tag{5.A.3}$$

This leads to

$$\mathbf{a} = (\mathbf{D}^t\mathbf{D})^{-1}\mathbf{D}^t\mathbf{c} \tag{5.A.4}$$

To extend to n-dimensional output, vector $\mathbf{c}$ is replaced by a $T \times n$ matrix $\mathbf{C}$. The foregoing analysis is applied to each column of $\mathbf{C}$, and the resulting linear transformation is now an $\mathbf{m} \times \mathbf{n}$ matrix $\mathbf{A}$ rather than vector $\mathbf{a}$. This results in Equation 5.14.

Direct inversion of the matrix $\mathbf{D}^t\mathbf{D}$ in Equation 5.A.4 can result in numerically unstable solutions, particularly if the system is noisy or ill conditioned. A more robust approach is to use singular value decomposition (SVD). A theorem in linear algebra states that any $T \times m$ matrix $\mathbf{D}$ of rank r can be represented by SVD, given by

$$\mathbf{D} = \mathbf{U}\Sigma\mathbf{V}^t \tag{5.A.5}$$

where $\mathbf{U}$ is a $T \times T$ unitary matrix whose columns $\mathbf{u}_1, \ldots, \mathbf{u}_T$ are the orthonormal eigenvectors of $\mathbf{D}^t\mathbf{D}$; $\mathbf{V}$ is an $m \times m$ unitary matrix whose columns $\mathbf{v}_1, \ldots, \mathbf{v}_m$ are the orthonormal eigenvectors of $\mathbf{D}\mathbf{D}^t$; and Σ is a $T \times m$ matrix given by

$$\sum = \begin{bmatrix} \Delta & 0 \\ 0 & 0 \end{bmatrix} \tag{5.A.6}$$

where Δ is a diagonal $r \times r$ matrix. Proof of this theorem, which is beyond the scope of this chapter, is found in Noble and Daniel.[93]

The diagonal entries, $\sigma_1, \ldots, \sigma_R$, of Δ are the singular values of $\mathbf{D}$. Equation 5.A.5 can be written in series form as

$$\boldsymbol{D} = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^t \tag{5.A.7}$$

Substituting Equation 5.A.7 into Equation 5.A.4, we get

$$\mathbf{a} = \sum_{i=1}^{r} \sigma_i^{-1} \mathbf{v}_i \mathbf{u}_i^t \mathbf{c} \tag{5.A.8}$$

An ill-conditioned or noisy system will result in some σ_i taking on very small values, thus resulting in an unstable solution to Equation 5.A.8. Singular value decomposition handles this situation gracefully. A stable least-squares solution can be arrived at by simply eliminating terms corresponding to very small σ_i in the summation of Equation 5.A.8.

A thorough formulation of SVD is given in Reference 93. C software for computing SVD and using it to solve the least-squares problem is provided in Chapter 2 of Reference 17. Popular mathematical packages such as Matlab™ also offer SVD based matrix inversion and least-squares solutions to linear systems.

appendix 5.B

Derivation of 3 × 3 matrix from display RGB to XYZ given white point and chromaticities of the primaries

Given chromaticity coordinates of the three primaries, $[x_R, y_R]$, $[x_G, y_G]$, $[x_B, y_B]$, and the tristimulus values of the white point, $[X_w, Y_w, Z_w]$, the goal is to derive the 3×3 matrix $\mathbf{A}_{CRT}$ that maps display RGB to XYZ as in Equation 5.55.

Assign an arbitrary value $Y_R = Y_G = Y_B = 1$ for the luminance of the three primaries. This provides three-dimensional xyY descriptors for the primaries.

Convert the xyY coordinates of each primary to XYZ space as follows:

$$X'_R = \frac{x_R}{y_R}, \quad Y'_R = 1, \quad Z'_R = \left(\frac{1 - x_R - y_R}{y_R}\right) = \frac{z_R}{y_R} \tag{5.B.1}$$

Analogous expressions apply for the green and blue primaries. This defines a matrix $\mathbf{A}'$ given by

$$\mathbf{A}' = \begin{bmatrix} X'_R & X'_G & X'_B \\ Y'_R & Y'_G & Y'_B \\ Z'_R & Z'_G & Z'_B \end{bmatrix} = \begin{bmatrix} x_R/y_R & x_G/y_G & x_B/y_B \\ 1 & 1 & 1 \\ z_R/y_R & z_G/y_G & z_B/y_B \end{bmatrix} \tag{5.B.2}$$

We now have to scale the column vectors of $\mathbf{A}'$ so that an input of RGB = [1, 1, 1] results in the desired white point $[X_w, Y_w, Z_w]$. This is done by the following operation:

$$\mathbf{A}_{CRT} = \mathbf{A}' * diag(\mathbf{A}'^{-1} * \mathbf{W}) \tag{5.B.3}$$

where **W** is the white vector.

chapter six

Digital color halftones

Charles Hains
Shen-Ge Wang
Keith Knox
Xerox Corporation

Contents

6.1 *Introduction*

Digital halftoning is a method of reducing the number or gray levels or colors in a digital image while maintaining the visual illusion that the image is still a continuous tone. This is done by trading spatial resolution for grayscale resolution. Black and white or single-color pixels are clustered together in groups so that, when viewed by the eye, they give the impression of gradation of color.

A halftone is needed to display an image on media that cannot reproduce many levels or colors. The prime example of this is the print media. With the exception of a few continuous-tone desktop printers, the vast majority of printed material uses halftones to represent images. Today, those images are almost exclusively prepared digitally, with digital halftones. Most computer monitors display images in color and continuous tone but, because of bandwidth limitations, the very highest resolutions are shown with reduced numbers of colors. To display an image on a monitor with reduced colors requires color quantization and digital halftoning. Color quantization is discussed in a separate chapter.

The goal of this chapter is to introduce the topic of digital color halftones. The focus will be on color halftones for printed material, with an emphasis on the problems and trade-offs that need to be made when choosing the type of halftoning method or structure to use for a given image or print medium.

From this chapter, one can learn to recognize the different types of halftone structures and understand where and when to use which digital color halftoning technologies.

6.1.1 History of halftoning

Halftoning had its start in photography. Before the first photograph was made by Nicéphore Niépce in 1826, images were printed either by carving an image on wood or metal. Intrigued by photography, William Henry Fox Talbot began experimenting with capturing images with light in 1833. Later that year, he developed the first negative–positive photographic process.

In an attempt to simplify the engraving of images on plates, Talbot developed the first halftoning method,[1] by recording an image directly on a printing plate. He used a black, cloth screen that he placed between the sensitized plate and the object to be reproduced, hence the term "screening." The weave of the cloth broke the image into small dots that varied with size, depending on the amount of light present in the image.

Improvements in halftone screens came slowly by modern standards. In 1893, Louis and Max Levy created a crossline screen[2] made of ruled lines on two pieces of glass that were rotated 90° with respect to each other and bonded together. The screen was placed in the camera a short distance away from the sensitized plate. Because the screen was slightly away from the plate, the exact position of the screen affected the tone reproduction of the process.

Sixty years later, a screen was developed that could be placed directly in contact with the sensitized plate.[3] The contact screen is a continuous-tone image of the pattern produced by the crossline screen when it is in the proper position. This removes the variability that is introduced by the position of the crossline screen. Contact screens became the mainstay of the printing industry until the advent of the digital halftone, at which point they became the model for the digital halftone clustered dot screen.

The origins of digital halftoning begin in the 1960s with the need to reproduce digital images on the early binary displays. Severe bandwidth limitations restricted how much raster data could be transmitted over data lines and displayed on monitors. The early work, in the 1960s, concentrated on methods of adding a noise signal, or dither signal,[4] to the image before quantizing. Later work, in the 1970s, shifted to periodic patterns, or ordered dither.[5] Analysis of these patterns,[6] showed that they were optimally distributed to minimize visibility of the halftone texture. Because the displays were of such low resolution and they did not have the instability problems of printers, dispersed texture patterns were acceptable and useful. Dispersed halftone screens are still in use today on low-resolution and low-bit-depth displays.

With the advent of laser and electronic printers in the 1970s, printer manufacturers turned to the question of replicating the traditional analog halftone screens. The early work on ordered dither recognized that replicating the analog screen[7] was one of the possible ways of arranging the ordered

dither signal. The higher resolution available on a printer made the use of these clustered dot screens feasible. Besides, as printers had known for 100 years, clustered dots, as in analog halftoning, were less susceptible to printer instabilities. As researchers worked through the decade, the problems of dot gain and tone reproduction[8] were analyzed and solved for digital halftones.

The invention of imagesetters in the late 1970s and early 1980s changed graphics arts. Now, type and images could be created together in raster form. Digital halftones of high resolution and high quality became an important topic of study. Of particular interest were the issues of how the different halftone screens interact in color printing. The initial digital halftone screens could not be generated at arbitrary angles, so the problems of color moiré became very important. This continues to be an area of current study. In this chapter, we will look into these issues and describe some of the latest efforts in this area of study in digital color halftones.

6.2 *Digital color halftones*

6.2.1 *Halftone structure*

Many different digital halftoning technologies have been being developed over the last 40 years. This overview will describe a few of them and will focus on the algorithms that a user is most likely to encounter. Two aspects of the halftoning algorithms will be emphasized: the nature of the halftone structure produced and the complexity of the implementation required to generate them.

Halftone structure is the most visible characteristic of the halftone algorithm. It is what the end user of the halftone sees. The quality of the image will be based on how pleasing the halftoned image looks to the eye. The two types of halftone structure are called *amplitude modulated (AM)* and *frequency modulated (FM).*

The AM halftone structure is named for the amplitude modulation encoding system used in telecommunications. A few levels of an AM halftone are shown in Figure 6.1. Amplitude modulation encodes a signal by modulating the amplitude of a carrier of fixed frequency. In the same way, the AM halftone produces a pattern of halftone dots with a fixed frequency, where the size of the dots varies with gray level.

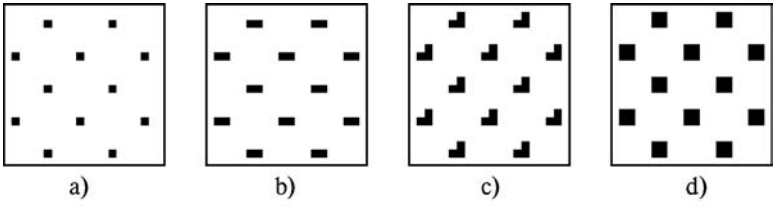


Figure 6.1 AM halftone structure of a 32–level clustered halftone dot.

The images shown in Figure 6.1 are the result of halftoning four different gray patches. Because the input images are uniform constants, the dot pattern itself can be studied. The images on the right side of the figure with the larger black dots correspond to darker gray levels. This halftone dot pattern is also called a *clustered halftone dot*, and it is analogous to the optical contact halftone screen.

In a similar manner, the FM halftone structure is named for the frequency modulation encoding system. An FM halftone structure is shown in Figure 6.2. Frequency modulation encodes a signal by changing the frequency of a carrier of fixed amplitude. The FM halftone structure is made of many pixels of the same size that are evenly dispersed, so the frequency of the pattern is a function of the gray level represented.

If a uniform input patch is halftoned with an FM halftone algorithm, the FM structure becomes visible. The input image levels are the same for both Figures 6.1 and 6.2. The more closely spaced black pixels correspond to darker gray levels, and widely spaced pixels to brighter gray levels. The dispersed pattern shown was generated by error diffusion, which is discussed in greater detail in Section 6.8. Error diffusion is an adaptive algorithm, so there is no fixed pattern for any given gray level.

6.2.2 *Threshold array halftone algorithms*

The complexity of the halftone algorithm implementation is an important factor in deciding when to use that algorithm. If it is a very simple implementation, then it can be used both in high-volume systems, operated in real-time, and in low-cost systems that cannot afford expensive, special-purpose hardware. A more complex implementation requires either specialized hardware or a very patient user.

The simplest and most common implementation halftone method is the threshold array. This is the method used to implement the most commonly used halftone algorithm, the clustered dot screen. The threshold array is very fast to execute and simple to implement. This makes it ideal for high-volume, high-resolution, real-time systems.

The threshold array is a two-dimensional array of numbers that defines the halftone cell. For the clustered dot, this array is a digital representation of the optical pattern used in the contact screen. In Figure 6.3, the halftone

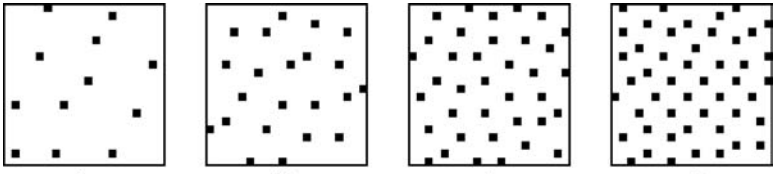


Figure 6.2 FM halftone structure generated by error diffusion.

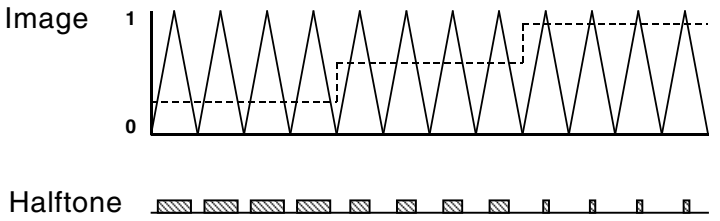


Figure 6.3 Threshold array halftone algorithm.

screen is shown in one dimension as a sawtooth pattern. The halftone cell is represented by a single tooth in that pattern. The cell is repeated across the image to create the complete halftone screen.

A one-dimensional image is shown superimposed on the halftone pattern. In this example, the image is shown as three different constant levels. As the cell is replicated across the image, each element in the cell is compared in value to the image pixel at that location. If the image pixel is larger than the threshold value at that location, then a white pixel is stored in the output array; otherwise, a black pixel is stored. This simple comparison will generate an array of black and white dots across the image. The size of the dots will vary, depending on the brightness of the image in that region. The darker that region of the image is, the larger the black dots will be. The spacing between the dots, or the frequency of the halftone screen, remains the same, independent of the gray level of the input image. This makes the clustered dot halftone screen an AM halftone algorithm.

Other halftone dot structures can be implemented with a threshold array as well. A stochastic screen is a threshold array that generates a much different-looking halftone structure than the clustered dot. The output of a stochastic screen is a random-looking arrangement of black and white pixels. An example can be seen in Figure 6.4. In a stochastic screened image, the width of the individual dots remains the same — one pixel wide. It is the density of dots that changes with gray level. In a bright region, the black pixels are widely spaced in a low-spatial-frequency pattern. In darker regions, the black pixels are closer together generating a higher spatial frequency pattern. This change in frequency with gray level makes the stochastic screen an FM halftone structure. More details about how to generate a stochastic screen structure are given in Section 6.8.9.

A dispersed dot is a third type of threshold array halftone dot. An example of its halftone structure can be seen in Figure 6.4. A dispersed dot[6] is a periodic, deterministic pattern that produces images with very high spatial frequencies. It can still be seen on color images produced for reduced-bit-depth displays. The thresholds in a dispersed dot are arranged so that the black and white pixels are dispersed as widely as possible throughout the pattern at each gray level. This change in spatial frequency as a function of gray level makes it an FM halftone structure.

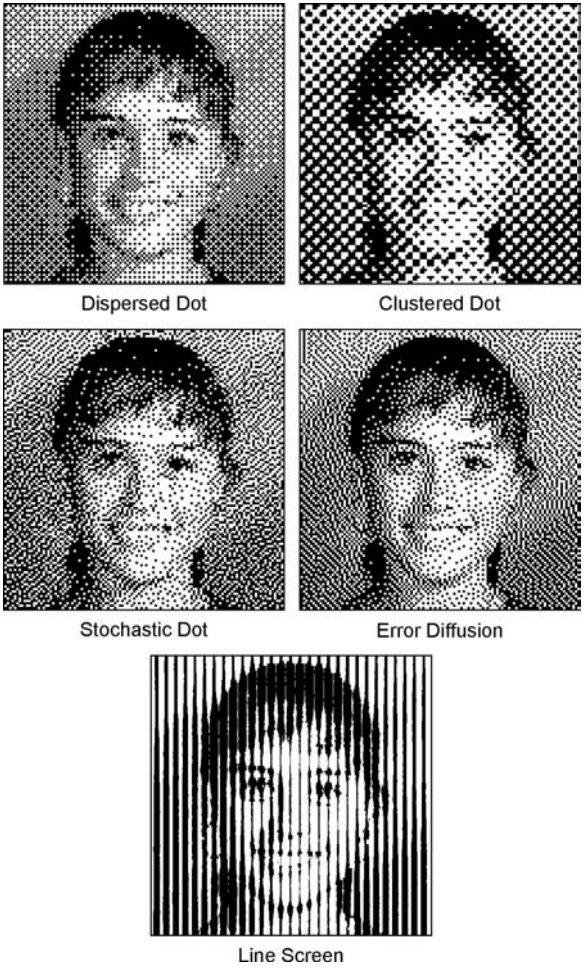


Figure 6.4 Comparison of clustered, dispersed, stochastic and line screens.

6.2.3 Spatially adaptive halftone algorithms

A spatially adaptive halftone algorithm is one that makes a decision about each pixel using image information from the neighborhood of that pixel. The algorithm may be single pass, multiple pass, or even iterative. All of this computation makes these algorithms very complex and computationally intensive. They are unlikely to be used on high-speed systems without specialized hardware assistance. To date, the majority of the algorithms using spatially adaptive methods generate an FM halftone structure. Some attempts have been made to modify these schemes to produce clustered dots of the type in AM halftones.[9,10]

Error diffusion was the first of the spatially adaptive algorithms. It is also the simplest. The algorithm is a single-pass method that computes the output value of a given pixel based on the value of the pixel and the results of all of the previously computed pixels. The details of the implementation method are given in Section 6.8.

The nature of the error diffusion algorithm causes it to generate collections of dispersed black and white pixels, with the spatial frequency of the patterns varying with gray level. This makes it an FM halftone structure. Another inherent feature of error diffusion is that the algorithm automatically edge enhances the input image. This enhancement is slight and directional, but it gives the impression that error diffusion reproduces detail better than other algorithms. Although the implementation of this method is more complex than the threshold array, its higher image quality has convinced some manufacturers to build specialized hardware to implement it.

The other class of algorithms that are spatially adaptive are the iterative algorithms. These algorithms involved a great deal more computation and generally are not used in commercial systems. They involve a many-pass, multiple-iteration calculation in which a cost function is evaluated, and the image is adjusted until equilibrium is reached. The nature of the image changes with the cost function chosen. Direct binary search[11] uses a visual filter to compare input and output images. Pulse density modulation[12] uses a criterion of maximum dispersion of the black and white pixels. Finally, spectral shaping[13] is used to adjust image quality with iterations between frequency and image space. All of these algorithms produce an FM halftone structure, are very compute intensive, and have to be performed individually for each image. The algorithms that are based on a human visual system (HVS) model are discussed in Chapter 7.

6.2.4 Trade-offs and color halftoning issues

The trade-offs and issues depend greatly on the intended output. Printer issues are different from those of displays, and different types of printers have different issues. There are some color issues that reach across all output devices — for example, the problems of color gamut and of registration of the color separations. Other issues such as dot gain, angled screens, and color moiré are specific to printing and even to clustered dot screens.

Dot gain is a term describing an increase in darkness of a halftone print as compared to what was intended by the binary digital pattern that was produced. Dot gain consists of two parts, physical dot gain and optical dot gain. Physical dot gain is due to the actual spreading of the ink or toner on the paper. The black dots in the printed image are larger than intended, and the image is darker. Optical dot gain, also called the Yule–Nielsen effect, is caused by the scattering of light in the paper. Light in the white parts of the paper, near the edges of the black halftone dots, will be scattered into the dot and be absorbed, thereby increasing the darkness of the printed halftone image. The effect of both dot gains is to make the size of the halftone dot

larger and the halftone darker. The increase in darkness due to dot gain can be measured, and the halftone dot can be calibrated to take the expected dot gain into account.

The more isolated black pixels there are in a halftone pattern, the more likely dot gain is to have an effect. This means that FM halftone structures, such as error diffusion, stochastic screens, and dispersed dot screens, are more susceptible to dot gain. Although calibration is possible, drifts in the printer response make these halftone algorithms difficult to keep stable.

The clustered dot halftone screen grows a dot by adding black pixels to the edge. The circular clustered pattern is the one shape that has the smallest circumference for a given area coverage. Because the ink and light spreading happens at the edges, the clustered dot screen is the least susceptible to the effect of dot gain.

Resolution can have an effect as well. A very high-resolution printing system may have difficulty holding single isolated black or white pixels. These pixels may just not print or may fill in. This makes FM halftone structures difficult to print when the resolution is very high. For this reason, high-resolution printers have a preference for clustered dot halftones. Low-resolution printing systems on desktop printers, on the other hand, often can print isolated pixels easily. Because the resolution is so low on the desktop printers, the high spatial frequency content of FM halftones is more desirable.

Registration of the three- or four-color separations, CMYK, can be an important issue. Depending on how the individual pixels in the four separations overlap, different colors are produced. This happens because the inks are not perfect, and they absorb in areas of the spectrum that interfere with the other colors.

The registration problem is solved with clustered dot screens by combining the screens together at different angles. The different screens beat with each other, causing the familiar rosette patterns. These rosettes are a moiré pattern that is created at a high enough spatial frequency to be acceptable. If the angles of the screens are not exactly right, then other moiré patterns will result that are unacceptable. These questions are discussed in detail in other parts of the chapter.

Registration can be a problem for FM halftone structures as well. There are no periodic screens in FM halftones, so moiré beat frequencies do not occur. There is random overlapping of isolated pixels between the color separations, however, that will cause color errors to occur.

In summary, clustered dot halftone screens are the digital emulation of the optical contact screens from the graphic arts. They work best in high-resolution printing systems, because their clustered dots are less susceptible to dot gain, and their color reproduction is straightforward to calibrate.

Frequency-modulated halftone systems, such as error diffusion, stochastic screens, and dispersed dots, work best in low-resolution systems such as desktop printers and displays. The low-resolution systems can reproduce the isolated pixels in the FM structures, and the high spatial frequencies of

the halftones are less visible on the low-resolution systems. Color reproduction is not as easy to keep stable, but it is not as important on these systems.

6.3 Halftone selection

This section is a "cookbook" approach to halftone selection. It presents step-by-step decisions for choosing the appropriate halftone screens for a new digital color printer. The discussion is primarily centered around laser-scanned xerographic printers; however, much of it applies to other devices as well. Some conclusions might differ for other technologies such as LED arrays or inkjet printing. Some of these differences are pointed out within the text.

6.3.1 Noting the printer's special characteristics

The first step is to analyze the digital printer to determine characteristics that may affect halftone performance. It is assumed that the printer is driven with a virtual raster image with orthogonal axes. For convenience, it is assumed that the horizontal axis is the fast-scan direction, and the vertical axis is the process direction. In the simplest case, the pixel frequency is the same in both directions. The pixel frequency is commonly called the *resolution* of the printer. This resolution should not be confused with actual optical resolution, because the virtual pixels in the raster image do not normally reach the paper without distortion (see "Dot gain" in Section 6.4.10.5).

It is not necessary for the pixel frequency to be the same in both directions. When this is the case, the base resolution is the direction that is most difficult to change. This would be the scan lines per inch in the case of a polygon scanning system, or the diode spacing on an LED bar system. It is usually the lower frequency of the two directions and is also closer to the actual optical resolution.

The other direction is easier to change and also is of higher frequency. It is related to the base resolution by what is called the *addressability* factor (see "High addressability" in Section 6.4.8). The addressability is usually an integer multiple of the base resolution such as 2:1, 4:1, or 8:1. For example, a polygon raster system of 600 lpi with a fast-scan frequency of 4800 pixels per inch would be called a 600-lpi printer with 8:1 addressability. Note that the scan line frequency is determined by the polygon and optics of the system, while the higher fast-scan frequency is achieved simply with an increase in the clock frequency along a scan line.

An LED bar system base resolution is fixed by the LED spacing, while the pixel frequency in the process direction is controlled by the LED clock frequency. So the LED-bar addressability is vertical, while the polygon scan addressability is horizontal.

If the addressability is not an integer multiple of the base resolution, it may complicate the design of the halftone screens.

Dot gain is one of the most important limiting factors for halftone printing. Dot gain is the extent to which halftone dots increase (or decrease) in

size from the digital pixel map that is created by the image processing (see "Dot gain" in Section 6.4.10.5). For an inkjet printer, the effective pixel will be round and generally much larger than the virtual pixel. High addressability in an inkjet printer can be effective but must be used in conjunction with an ink-reduction algorithm.

For a polygon laser or LED-bar scanning system where the effective beam is Gaussian in profile, high addressability can be very effective in increasing the amount of information that can be delivered to the photoreceptor and thus the number of levels in a digital halftone screen. This is true even though the optical resolution has not been changed with the increased addressability. The use of an oversized Gaussian beam will add to the dot gain in such a system. While the overall dot gain may be positive in a xerographic printer, high addressability may result in some negative dot gain in isolated high-addressability pixels if they are very narrow with respect to the beam size.

Note whether the printer has *multilevel writing* capability or is inherently a *binary* system. Thermal head, dye diffusion, and multi-drop inkjet printers can be true multilevel writing systems. They may not need halftoning at all or may use a hybrid halftone.

Xerographic or electrographic laser exposure printers are generally binary. Even when multilevel writing is claimed, the intermediate levels may be less stable and may be useful only for smoothing the periphery of otherwise binary halftone dots. Some claimed multilevel writing is actually achieved with high addressability.

The *bit-depth* that is carried through the image processing is generally eight bits (256 values) per separation. This is a good match for what the eye can see, what the printer can render, and the cost of image processing circuitry. Many scanners deliver more than eight bits per separation, which may be necessary in the early stages of image capture, especially when the scanner works in reflectivity space. The conversion to units that are closer to human visual will have some quantization loss. But after initial processing, eight bits is generally sufficient. If higher bit depth is actually available, it may be possible to increase the final image quality, but at a considerable increase in complexity.

The *repeatability* or reproducibility of the printer must be determined — in particular, the misregistration that is produced. The amount of misregistration determines whether rotated halftone screens are necessary or if dot-on-dot or line-on-line printing might be possible. The misregistration also affects the requirement for trapping in the image processing stage.

The *noise* level of the printer may affect the halftoning strategy. Image noise sources include polygon jitter, polygon signature, paper speed variations, banding, streaking, colorant granularity, mottle, and LED variability. The amount of noise may limit the choices of halftone frequencies to be used. Some screen angles are more sensitive to printer noise — in particular, 0 and 45° screens.

Interactions between color separations can limit halftoning choices. Ideally, the color separations should not affect each other. It must be determined whether the presence of one separation affects the laydown or the transfer of another separation. In some conventional ink printing systems, the laydown of a subsequent ink layer may actually remove some of the earlier ink. In some of the new intermediate-transfer xerographic printers, the presence of toner from earlier separations on the intermediate belt or drum may hamper transfer of new toner at some halftone frequencies. This could be considered to be a *mechanical moiré* rather than an optical moiré.

The color characteristics of the *colorants* (inks or toners) can have a big effect on image quality. To have process color capability, the cyan, magenta, and yellow colorants must be transparent. To the extent that they lack transparency, image color quality will be compromised. The spectral characteristics of the colorants are also most important. The size of the printer color gamut is determined by the spectral characteristics. And much of the color moiré comes from the unwanted absorption characteristics of the colorants. The colorants used will also affect the *under-color removal/gray-component replacement (UCR/GCR)* strategy and thus the screen angles and frequencies that can be used.

6.3.2 *Decisions involved in choosing halftone structure*

After considering the characteristics of the target printer, the next step is to choose the basic halftone structure. The choices discussed below include:

- Continuous tone
- Line screens
- Clustered dots
- Dispersed screens (periodic)
- Stochastic screens
- Error diffusion

Figure 6.4 (page 392) shows a comparison of five of these screen types.

If *continuous tone* (contone) is really an option, it may be preferred, as it can avoid most of the color moiré problems as well as the visibility of color rosettes that result from rotated-screen printing. But if there is not enough dynamic range for each color (i.e., not enough gray levels), then it may be necessary to use a combination of contone with halftones to get enough colors. In this case, the contone levels might be used to "soften" the edges of the halftone dots.

Recent developments in color copiers have used a method that has been called "contone" but is in fact a very fine-line screen (200–600 lpi). They also use a "line-on-line" halftoning orientation that is unconventional in the printing industry. There are several reasons why these methods might work for a copier but should be avoided in a desktop publishing situation. To begin with, copier output is not expected to be high quality. Originals to be

copied are often of low quality and usually contain halftones that cause difficulty in copying. Also, the "contone" line screen is a good match for the relatively noisy xerographic process. The noise in the process appears to mask or avoid the color instabilities that would occur if the same process were used in conventional printing.[14] Last, the copier is producing the final print that is to be viewed. If sharp photo-ready images or printing plates or films were the object, then this method might not be appropriate.

Clustered halftone dot screens are usually successful in digital binary printing systems for reasons that are similar to why the conventional printing industry uses them. The largest driving force is printer dot gain. If the printer can produce a very reliable isolated printer pixel, then other strategies can be considered, but clustered dots have the capability to calibrate out the dot gain problem and limit the problem to the extreme highlights and shadow regions.

Dispersed screens (periodic) are commonly seen on computer displays, especially in website images. They are effective for several reasons; first, the display resolution is low so that a clustered dot would be unacceptably coarse. And, second, the display has virtually no dot gain. When the printer has a very dependable isolated printer pixel, such as produced by an inkjet printer, then dispersed screens can be considered. At lower resolutions, a dispersed screen may be less objectionable than a stochastic screen or error diffusion.

Stochastic screens (dispersed nonperiodic), including those called "blue noise masks," have become more popular but should be used with caution on a laser-scanned printer. They are desirable because they can avoid color moiré problems and subject moiré problems associated with cluster-dot printing. They also can provide greater image detail and sharpness as well as gradations without contours. However, they add a graininess or visible noise to an image, and they are very difficult to calibrate and proof. And, without a stable, repeatable printer pixel, they have extreme dot gain problems throughout the grayscale range.

The use of stochastic screens is becoming popular in digital printing systems, with mixed success. It is true that the stochastic screen is an important tool that should be considered as a supplement to clustered dot screening. Stochastic screens:

- Can eliminate subject moiré as well as color moiré
- Can provide greater image detail and sharpness
- Can provide smoother gradations without contours
- Can provide more latitude on the press for inking and pressure
- Are especially effective for low resolution

However, they come with a down side, as stochastic screens:

- Can add graininess or visible noise to an image
- Require stable, repeatable printer pixels

- Have extreme dot gain problems across the grayscale range
- Are very difficult to proof, especially on a different printer
- Require cooperation between the creator and the printer
- Require sophisticated area-coverage models with knowledge of neighboring pixels in all colorants to provide good color correction

In the 1980s, industry leader Dusty Rhodes said, "The success of electronic publishing will require the industry to evolve from the state where the printer is an artist into the state where the printer is a technician." For this evolution, desktop and demand printing on remote devices will require the use of printer-independent files and devices that are consistent and predictable. The printer will need to maintain the devices in a nominal calibrated condition.

It would seem that the growing use of stochastic screens in high-end printing applications is a return to the state where the printer is more of an artist. See more about stochastic screens in Sections 6.7 and 6.8.

Error diffusion is a computationally intensive screening method that avoids quantization errors by distributing errors into local neighborhoods. It is most effective at low and intermediate resolutions and for computer display screens. It gives the appearance of sharpness and eliminates contouring from insufficient levels. There is some edge enhancement inherent the in algorithm.[15] Error diffusion can be very effective for inkjet printing. However, it is nearly impossible to calibrate, especially in a printer with significant dot gain. It is possible to use error diffusion techniques not for binary halftoning but as a gray-to-gray operation in conjunction with other halftoning methods.

In short, dispersed screens such as stochastic and error diffusion can be important supplemental tools but may not produce the highest color image quality in a laser-scanned or LED-driven printer. In the inkjet world, they provide the best image quality and are therefore widely used.

6.3.3 Choosing frequencies, stochastic, cluster, error diffusion

After choosing the basic halftone structure, the next step is to choose the appropriate size. For cluster dots, dispersed screens, and line screens, this means choosing the screen frequency. For stochastic screens, this means choosing the cell size (see Table 6.1). For error diffusion, it may mean choosing the smallest spot size.

Experimentation with the target printer is the best way to determine these values. A number of potential screens should be designed and tested. These should include screens that are both higher as well as lower frequency than the anticipated screens. The final screen is generally a compromise that trades off higher frequency against printer defects, and it cannot be determined without considering all the possibilities.

The test screens do not need to be well designed or tuned. The quality indicator to look for at this stage is whether the printer can hold the frequency.

Table 6.1 Cluster Dot Cell Size

Characteristic	Large Dots	Small Dots
False contouring	Less	More
Stability/uniformity	More	Less
Frequency response	Lower	Higher
Visibility/noticeability	Higher	Lower
Optical dot gain	Lower	Higher
Available angles	More	Fewer
Text performance	Poor	Better
Subject moiré	Better	Poor

Poorly designed dots may result in contours in an image due to jumps in effective density from level to level. This problem can be improved later with halftone tuning.

The test image should include a diagonal wedge or gradation that sweeps from white to black. Both horizontal and vertical wedges can be ineffective, because certain printer problems can mask the performance of the halftone screens. The diagonal sweep allows the eye to separate printer problems from halftone problems. Printer problems such as banding and streaking tend to be horizontal or vertical in appearance.

These test prints will show the trade-off that needs to be made. Halftones with too low a frequency will have visible dots and may break up small text and fine graphics. Halftones with too high a frequency may be more sensitive to printer noise and may not have enough levels. High-frequency dots may also be less robust and more difficult to calibrate.

6.3.4 *UCR/GCR strategy: minimum or maximum*

Before investing heavily in detailed dot design and fine tuning, some consideration should be given to the black-printer strategy. The black-printer strategy, sometimes referred to as *under-color removal (UCR)* or *gray component replacement (GCR),* includes the degree to which the black colorant carries the neutral information.

In a three-colorant cyan–magenta–yellow printer, there is no black colorant. The neutral colors are produced by relatively equal combinations of the three colorants. The four-colorant printer includes black as well as the other three colorants.

In a strong UCR/GCR strategy, there is a heavy reliance on the black colorant for all neutral colors. In the extreme 100% UCR case, it is possible to ensure that all colors are made up of at most three colorants and that cyan, magenta, and yellow do not occur simultaneously at any point on the print.

In a skeleton black strategy, the black colorant is used primarily to extend the color gamut in the shadows. This is often necessary when the three-color

combination of cyan, magenta, and yellow is not dense enough or is not close enough to neutral. The black colorant can be used to correct the color deficiencies of the three-colorant system. There can be a continuum of strategies among these extremes.

For the strong UCR/GCR strategy, the black print is much more apparent and requires more attention to the black screen design. This strategy also results in more chances for four-color moiré throughout the color range, so more attention must be given to the screen angles for the four-dot set.

For the weak UCR/GCR strategy, the black print is much less apparent, especially in the highlights and midtones. And there are fewer chances for four-color moiré. Less attention may be required to the black screen design, but more attention may be required for the other colors, as there are increased chances for two-color moiré to occur.

Early testing for moiré may indicate that one of the UCR/GCR strategies might be a requirement rather than an option.

6.3.5 *Choosing family of angles*

In digital color printing, the color separations are halftoned and imaged independently. Because of this, there are many chances for very slight misregistration to occur from one separation to the next. Misregistration of halftone dots can produce unpredictable colors because of the overlap of colorants that have unwanted absorption in their spectra. Rotated halftone screens are normally used to gain insensitivity to misregistration by randomizing the overlaps.

The conventional strategy is to choose equally spaced angles for the more visible screens of black, cyan, and magenta, and to insert the less visible yellow screen between two of the others, e.g., 45° black, 15° magenta, 75° cyan, and 0° yellow. This is the best strategy in most cases. In cases of minimum UCR/GCR, it may be better to use the good 45° screen for cyan or magenta instead of black. In cases of severe two-color moiré, it may be desirable to use a nonclustered screen for the yellow (see "Moiré considerations" in Section 6.4.12). Rotated screen printing produces a rosette pattern that, at best, is about twice as large as the individual screen pattern. Accurate angles are required to avoid low-frequency beats with even larger patterns.

6.4 *Clustered halftone screen design*

6.4.1 *Implementation*

Digital halftoning requires comparing input image data at each output printer pixel with a halftoning function. This can be done in several different ways: by adding the function to the image data and thresholding the sum, by subtracting the function from the data and testing the sign, or by table look up. Figure 6.5 shows schematically the subtraction of the function from the data and the resultant binary output based on the sign.

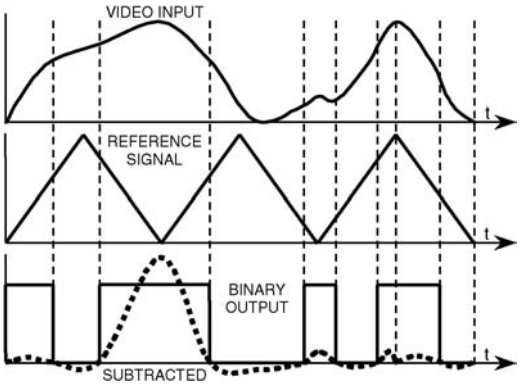


Figure 6.5 Digital halftoning waveforms.

Note in this figure that is possible to have some detail in the resulting binary output that is actually at a higher frequency than the halftoning function. This ability is called *partial dots* and is the result of making a separate comparison for every printer pixel rather than a single comparison for a screen cell (see "Partial dots" in Section 6.4.3).

It is only necessary to store a single halftone cell during processing and to step through the cell sequentially and repeatedly so as to create a virtual full-page halftoning function. The individual cells must "tile" together in order to tessellate the plane.

Figure 6.6 shows how a simple zero-degree halftone cell tiles together to produce a uniform function. The figure also shows how a digital halftone produces grayscale by modulating the area coverage of the colorant one pixel at a time. It also hints at the basic trade-off in digital halftones: number of levels vs. screen frequency. A larger cell has more pixels to turn on sequentially

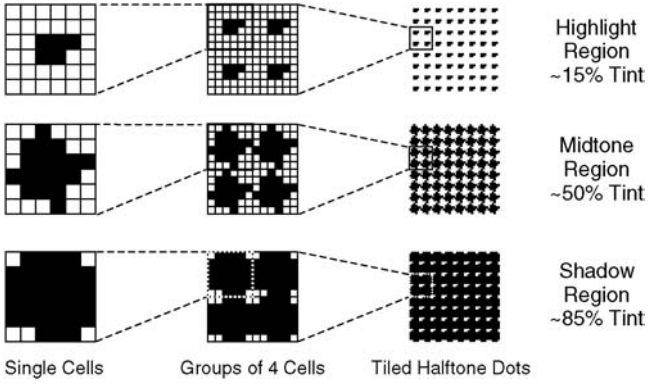


Figure 6.6 Tiling with halftone cells.

and thus has more possible gray levels. However, the larger cell replicates at a lower visual frequency.

6.4.2 *Spot functions, threshold arrays, table look-up*

The spot function in PostScript™ was an early digital method of describing the halftoning function. A spot function is a mathematical description of a zero-degree screen. The screen is rotated to an approximation of the desired angle in real time during processing. The method is susceptible to integer-arithmetic roundoff and quantizations during processing. For high-end imagesetters with high resolution, the method was adequate. No fine tuning of the screen is possible with a spot function, so it is not as effective for a mid-resolution printer where small errors can become very visible.

The threshold array method utilizes an array of threshold values that are stepped through sequentially along with the image values, and a comparison is made on a pixel-by-pixel basis. The threshold array is generally rectangular and most efficiently has the same number of pixel thresholds as a single halftone cell. A PostScript™ Type 3 threshold array utilizes a single rectangular array, whereas an Postscript™ Type 10 array utilizes two square arrays rather than a singular rectangular array. While the threshold array method requires the least amount of storage, it has the limitation that the halftone dot growth must be monotonic; i.e., once a pixel is turned on in the growth sequence, it remains on for the remainder of the growth sequence.

The table lookup method makes all the threshold comparisons during screen design and stores the actual binary halftone output in a large array that is addressed by both the data value as well as phase indexing information. This method can require as much as 32 times as much memory, but it has no monotonic growth requirement. Each halftone level can be independent of the others.

Dispersed and stochastic screens (including blue-noise masks) are usually implemented as large threshold arrays in the same manner as clustered dots. An error diffusion screen, however, is the result of a two-dimensional neighborhood calculation and does not repeat on a regular basis.

6.4.3 *Partial dots*

One of the most important aspects of any digital halftone method is the requirement for partial dots. *Partial dots* refers to the ability of the halftone image to contain detail at greater resolution than the halftone screen mesh. This ability produces halftone dots that can be irregular in shape, hence the name *partial dots.* To achieve this effect, one sub-element of a halftone cell is compared with one pixel of the image to generate one bit in the output image. Each pixel of the image may, of course, be used more than once if the image is magnified by replication. The use of a partial-dot technique allows the high-contrast pixel resolution to be reproduced in the halftone image as a sub-element of the halftone dot vs. the whole halftone dot for

non-partial-dot techniques. This results in a dramatic improvement in detail of an image displayed or printed on any system with a limited number of bits across its width. A comparison of these two techniques, including examples, was made by Roetling.[16] Figure 6.7 shows the effect of partial dots.

Note that, along the high-contrast edges in the figure, many of the pixels line up to produce image detail. If the halftoning were instead done on a cell-by-cell basis like a "font," then very coarse jaggies would be seen along the edges.

Partial dots are achieved by making quantization decisions at the resolution frequency of the basic printer raster matrix rather than one decision per input sample. It is known that high-resolution and high-contrast information, such as solid text or line art, can pass through a partial dot halftone process without any modification, while a different process might produce "jaggies" at the resolution of the halftone. Note in the figure that many of the pixels line up to produce image detail along the high-contrast edges.

Figure 6.8 is an example of a simple halftone using partial dots. The inset picture is the halftone printed at normal resolution. The main figure is a

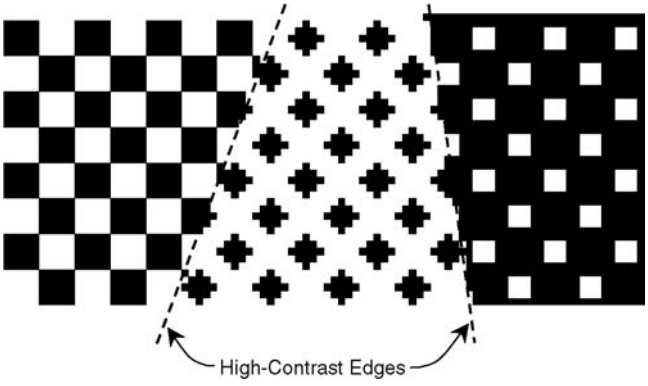


Figure 6.7 Partial dots.

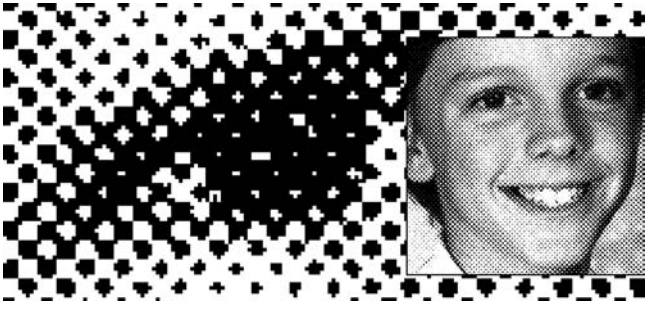


Figure 6.8 Example of simple halftone with partial dots.

blow-up of the child's eye. It can be seen in the inset that there is a strong impression of a circular pupil in the eye. The blow-up shows that impression is created by the presence of partial dot detail along the high-contrast edge of the pupil.

Partial dots are very important for image quality in digital printing, and every halftoning implementation should use them.

6.4.4 *Dot growth*

Figure 6.9 shows a typical dot growth sequence for a digital dot. It starts with a single printer pixel turned on in each cell; for each succeeding level, another pixel in the cell is turned on.

Note that there is a phase shift at the 50% growth point. In the highlights, the dot is a growing spot of colorant. In the shadows, the dot is a shrinking spot of white paper or a hole. At the 50% point, where the dots touch on all corners, the pattern shifts from spots to holes with a change of phase. Note also that, at the 50% point, the pattern forms a perfect checkerboard. This is important and is the only way that the screen can progress from circular spots to circular holes in a symmetric manner.

6.4.5 *Angled screens (Holladay)*

Digital halftone screens normally have been implemented by storing a matrix of screen thresholds in computer memory and comparing these thresholds with the image data on a pixel-by-pixel basis at printer resolution. This method is efficient and fast for simple 0° screens and 45° screens, but some implementations require enormous storage and complexity to achieve angles approaching the highly desirable ±15°. Holladay published a very efficient implementation technique for rational-tangent screens in 1980.[17] Rational-tangent screens, where the halftone dots are synchronized with the digital raster pitch, were the best software implementation until recent developments. However, rational-tangent screens have an inherent limitation in achieving all angles.

The Holladay halftone algorithm, as in most digital halftoning schemes, compares a stored halftoning function with picture values to create a one-

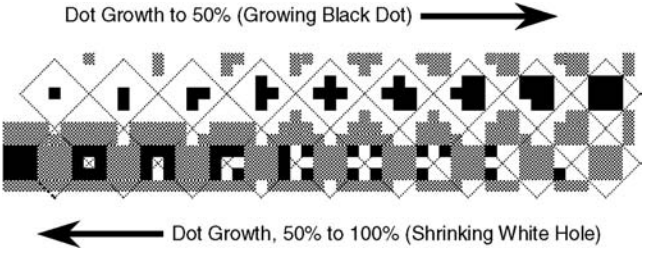


Figure 6.9 Digital dot growth sequence.

bit raster signal at printer resolution. The unique features of the Holladay technique include its ability to describe exactly the halftoning function in an efficient form for storage and its simplicity for either software or hardware implementation.

Most digital halftone algorithms for angled screens are notoriously inefficient in either one or all of the characteristics of speed, halftone screen storage, and output buffer space for the halftone image. Earlier techniques stored either a full page or a square section of the screen function that contained an integer number of cycles, which could become unwieldy for screens of arbitrary angles. The Holladay halftone technique requires storage for only one cycle of the halftone screen function regardless of angle, is very efficient in processing speed, and requires no output buffer. A more complete description of the Holladay technique can be found in the *Proceedings of the Society for Information Display* article titled, "An Optimum Algorithm for Halftone Generation for Displays and Hard Copies."[17]

Another benefit of the Holladay scheme is that the actual halftoning function is described in storage for any given angle of rotation and number of levels. This same function is used for each and every halftone cell, with very predictable results. With each dot being identical, there is no low-frequency moiré pattern introduced. In addition, these consistent dot patterns can be reliably measured and calibrated. Because the effect of every sub-element can be measured, the final version of halftone dot designs can be "hand tuned."

It is also possible, by assigning thresholds accordingly, to effectively linearize the printed response of these halftone levels with respect to the input grayscale sample data space. The calibration procedure can take into account the dot gain of the lithographic printing process as well, if measurements that are made on test patches are carried through from digital creation to plate creation.

The insight that Holladay had was that for a uniform tessellating pattern, any boundary drawn around a single repeating cell is completely arbitrary. In Figure 6.10, the repeating visual cell with 20 elements (A through L and 0 through 9) forms a pattern at 64°. But the figure shows that the arbitrary boundary can more easily be drawn as a simple rectangle with the same number of elements.

The resulting Holladay brick is much easier to implement in a digital system. It repeats horizontally across the page and has a horizontal shift for each successive row of bricks.

In general, there are three cases for Holladay dots. Figure 6.11 shows that the 0° screen is the simplest case with a repeating square brick that abuts on the corners without any shift.

The 45° screen is made from a 2:1 rectangle that shifts by half of its width for each successive row of bricks.

The third case is for the other angles. It is usually a long and thin brick with an odd shift. The figure shows the visual effect of the screens on the left and the implementation bricks on the right.

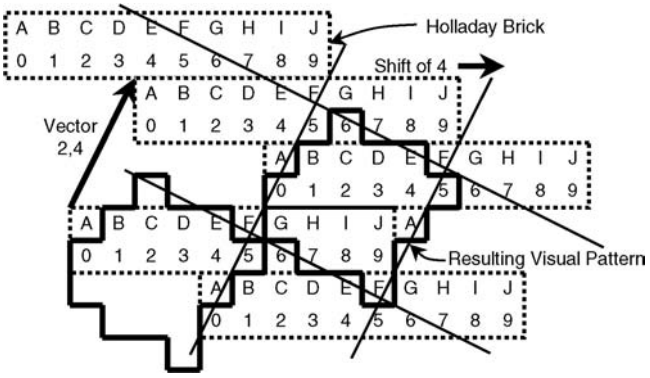


Figure 6.10 Holladay halftone geometry.

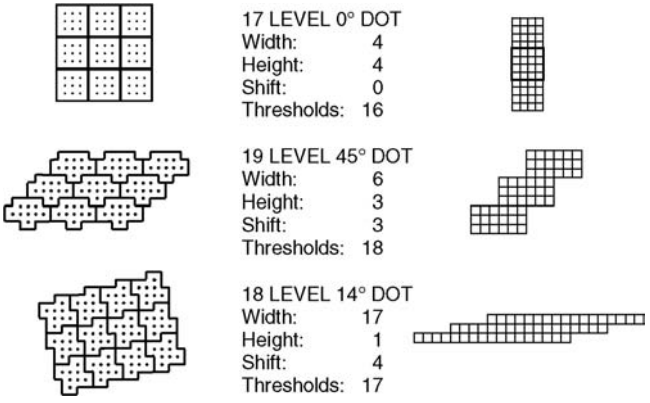


Figure 6.11 Holladay halftone examples.

6.4.6 Rational tangent screens

A limitation of a digital raster scheme is that not all rotation angles and screen frequencies can be produced exactly, although most can be closely approximated. For example, both the 0° screen and the 45° screen can be achieved exactly, but a 15° screen is usually approximated with perhaps a 14.04° or 18.43° screen. It should be noted that the screen angles that cannot be achieved easily in the Holladay scheme are the same angles that would produce the most objectionable moiré and low-frequency patterns in other approaches.

The dots that can be realized in the Holladay halftoning method can be categorized as vectors with integer values of x and y. There is only a finite set of such vectors. It is known that only patterns with this characteristic have the capability to tessellate a plane surface in the manner required by

digital halftoning. Because the angle that is achieved is the arc tangent of the integers y/x, the screens are referred to as *rational tangent screens.*

Figure 6.12 shows that for a vector of 6, 2, an 18° pattern is produced with 41 levels of gray. The visual patterns are identical, and the corners of the cells line up on grid intersections.

Table 6.2 shows all possible square dots up to levels of 250 and the associated angles for all combinations of integers of x and y up to 15. The notation "5/26.6" refers to 5 levels at an angle of 26.6°. Angles are rounded off to the nearest tenth of a degree. Only dots with angles between 0 and 45° are included. Others can be produced by interchanging x and y or making mirror images of the dots horizontally and/or vertically.

It can be seen that these dots have no reference to resolution or frequency but contain only screen angle and number of levels. It is a fundamental maxim for digital halftones that the balance between resolution and grayscale levels is a classic trade-off, similar to a gain-bandwidth product. Because the grayscale levels are produced by filling in a matrix of subelements for each halftone dot, increasing halftone levels implies lower frequency of the basic halftone screen mesh, and vice versa. A choice must be made between spatial resolution and perceivable grayscale.

In addition, because only integer vectors can be realized, the achievable ranges for resolution and grayscale are actually quantized into discrete choices. Figure 6.13 includes a plot of grayscale levels vs. dot frequency for various printer resolutions. Such plots are invaluable in choosing digital dots for a printing device.

From these figures, it can be seen that, for a given printer resolution choice and desired halftone screen frequency, a family of dots can be found with each of 0, 15, 30, and 45° angles approximated. It is not possible for

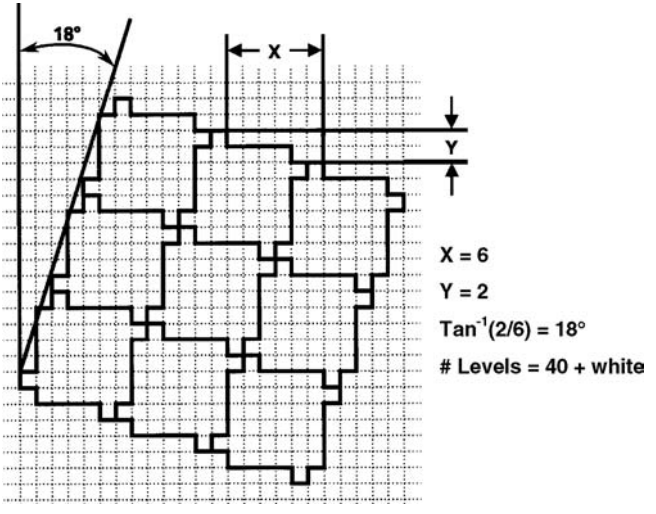


Figure 6.12 Rational tangent screen example.

Table 6.2 Available Rational-Tangent Screens

y	x = 2	3	4	5	6	7	8	9
0	4/0°	9/0°	16/0°	25/0°	36/0°	49/0°	64/0°	81/0°
1	5/26.6°	10/18.4°	17/14.0°	26/11.3°	37/9.5°	50/8.1°	65/7.1°	82/6.3°
2	8/45°	13/33.7°	20/26.6°	29/21.8°	40/18.4°	53/15.9°	68/14.0°	85/12.5°
3		18/45°	25/36.9°	34/31.0°	45/26.6°	58/23.2°	73/20.6°	90/18.4°
4			32/45°	41/38.7°	52/33.7°	65/29.7°	80/26.6°	97/24.0°
5				50/45°	61/39.8°	74/35.5°	89/32.0°	106/29.1°
6					72/45°	85/40.6°	100/36.9°	117/33.7°
7						98/45°	113/41.2°	130/37.9°
8							128/45°	145/41.6°
9								162/45°

y	x = 10	11	12	13	14	15	16	17
0	100/0°	121/0°	144/0°	169/0°	196/0°	225/0°	256/0°	289/0°
1	101/5.7°	122/5.2°	145/4.8°	170/4.4°	197/4.1°	226/3.8°	257/3.6°	290/3.4°
2	104/11.3°	125/10.3°	148/9.5°	173/8.7°	200/8.1°	229/7.6°	260/7.1°	293/6.7°
3	109/16.7°	130/15.3°	153/14.0°	178/13.0°	205/12.1°	234/11.3°	265/10.6°	298/10.0°
4	116/21.8°	137/20.0°	160/18°	185/17.1°	212/15.9°	241/14.9°	272/14.0°	305/13.2°
5	125/26.6°	146/24.4°	169/22.6°	194/21.0°	221/19.7°	250/18.4°	281/17.4°	314/16.4°
6	136/31.0°	157/28.6°	180/26.6°	205/24.8°	232/23.2°	261/21.8°	292/20.6°	325/19.4°
7	149/35.0°	170/32.5°	193/30.3°	218/28.3°	245/26.6°	274/25.0°	305/23.6°	338/22.4°
8	164/38.7°	185/36.0°	208/33.7°	233/31.6°	260/29.7°	289/28.1°	320/26.6°	353/25.2°
9	181/42.0°	202/39.3°	225/36.9°	250/34.7°	277/32.7°	306/31.0°	337/29.4°	370/27.9°
10	200/45°	221/42.3°	244/39.8°	269/37.6°	296/35.5°	325/33.7°	356/32.0°	389/30.5°
11		242/45°	265/42.5°	290/40.2°	317/38.2°	346/36.3°		
12			288/45°	313/42.7°	340/40.6°	369/38.7°		

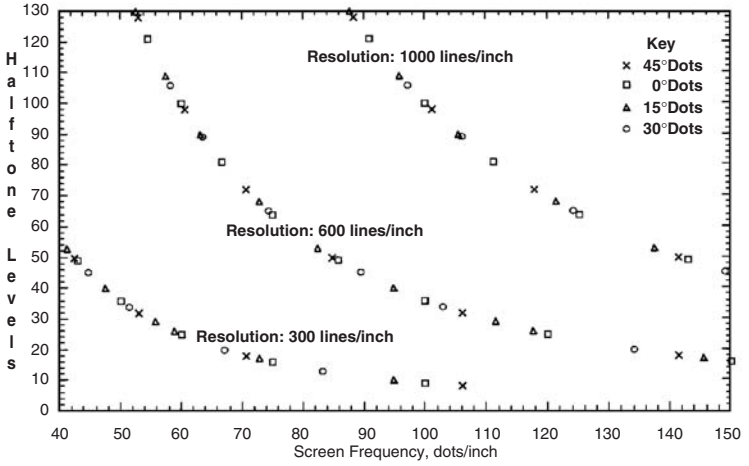


Figure 6.13 Trade-off of grayscale vs. frequency.

each of the dots to have exactly the same frequency or number of gray levels and to still have the simplicity, precision, and control associated with the Holladay scheme.

Figure 6.14 shows the simple mathematics for rational tangent screens. The parameters of dot area, printer resolution, and screen frequency are all interrelated. It is not necessary to draw a boundary or count the pixels. If you know the rational tangent vector and the printer resolution, then you can easily calculate the screen angle, the frequency, and the dot area (and the number of levels). Any one of the parameters can be computed from the others.

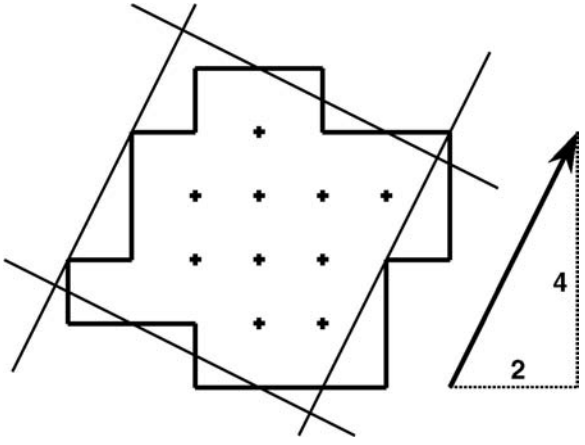


Figure 6.14 Halftone parameter equations.

For the example with a vector of 2, 4:

- *Screen angle* is the arc tangent of the vector x, y,

$$\alpha = \arctan(y/x) = 63.4°$$

- *Cell area* in pixels is the square of the vector length,

$$\text{Area} = x^2 + y^2 = 20$$

- *Grayscale levels* is the area in pixels plus one,

$$\text{Levels} = \text{Area} + 1 = 21$$

- *Vector length* is the square root of the cell area.
- *Frequency* is printer resolution divided by the vector length,

$$\text{Frequency} = \text{Resolution} \div \sqrt{\text{Area}} = 600 \div \sqrt{20} = 134 \text{ dpi}$$

- *Grayscale levels* is one plus the square of the resolution divided by frequency,

$$\text{Levels} = 1 + (\text{Resolution}/\text{Frequency})^2$$

- *Required resolution* is the frequency times the square root of the area,

$$\text{Resolution} = \text{Frequency} * \sqrt{\text{Area}}$$

6.4.7 *Supercells and accurate screens*

High-end digital scanners have solved the angle problem with sophisticated high-speed hardware. Dot-growth models can be electronically rotated in real time in hardware. Recent software approaches have produced near-perfect angles by the use of very large "supercells" (see Figure 6.15), which in reality are still rational-tangent but contain enough individual halftone cells to distribute the error over a large area. This is in effect the same thing that the high-speed hardware has achieved. These methods work well at the high resolutions that have been employed in expensive imagesetters and scanners in the past. When there are enough printer pixels in each halftone cell (in the hundreds), it is easy to achieve the desired dot shape, frequency, and angle with a simple algorithm. It is with lower-resolution desktop printers that the methods fail. Printers utilizing resolution from 400 to as high as 1000 lpi produce dots with less than 64 printer pixels. Here, the resulting difference from halftone cell to cell is visually significant. Lower-resolution devices require carefully "tuned" dot growth patterns to avoid visual patterns.

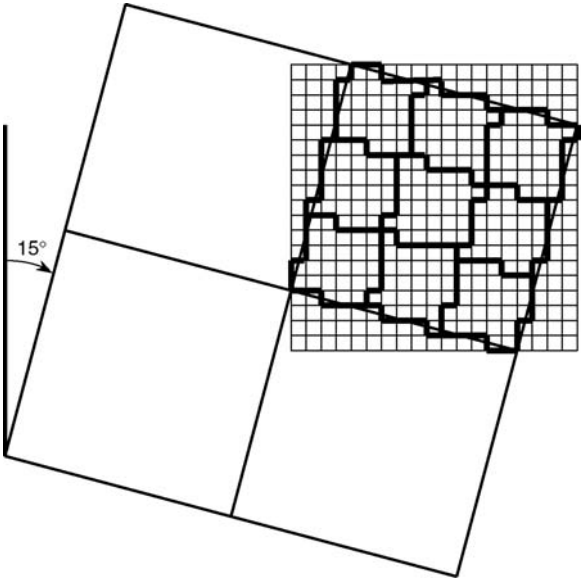


Figure 6.15 Supercell halftones.

Figure 6.15 shows that the individual cells that make up a supercell are not necessarily congruent. Each may have a slightly different shape and may contain a slightly different number of pixels. It is only in the collective average that the screen produces the desired effect. The figure shows that only the corners of the larger array of cells need line up with grid points. It is these longer grid point vectors that determine the angle — still a rational-tangent angle but much more precise.

6.4.7.1 Multi-center screens

Mid-resolution printers have achieved better angles and frequencies with smaller multi-center dots (2-, 4-, and as high as 10-center dots) that were carefully hand tuned, but some visual low-frequency artifacts remain.

A two-center dot, also called a *dual dot,* is common, because it is the easiest way to implement a 45° screen. The dual 45° dot is implemented in a 0° array that usually contains congruent cells that turn on in exactly the same sequence. In this case, it is used simply to avoid the step-and-shift of an angled screen and cannot be distinguished visually from a simple single-cell 45° screen.

But it is not necessary for the individual cells to contain the same redundant thresholds. The major goal in designing halftones is to achieve as many gray levels as possible so as to avoid visual contours in images. By staggering the thresholds in a multi-center dot so that the pixels turn on alternately in one center and then the other, the total number of gray levels becomes one plus the total number of pixels in the larger implementation array rather

than a single cell. This is a trick for the human eye: the visual frequency is produced by the smaller sub-cells, but the gray levels are produced on the average over the larger array.

Figure 6.16 shows a four-center, or *quad*, dot with a vector of 11, 3. In this case, we achieve two objectives by means of the multi-center dot; the average angle for the resulting screen is much closer to 15° than would be possible with the simple screen, and the dot has four times the gray levels. The downside is that some visual artifacts are also produced, particularly in the highlights and the shadows. The figure shows the first level turned on in the upper left quadrant. Pixels in the other three cells are not yet turned on. But also shown is this same first level turned on in adjacent copies of the tiled quad dot. It can be seen that this first level produces a pattern that is only one-half the desired frequency of the simple screen. When the next level turns on in another cell, another undesired frequency pattern is created. This happens again for the third level. It is not until the fourth level turns on that desired frequency is achieved. This situation repeats throughout the dot growth for the screen, although the undesired patterns are much less noticeable as the dot approaches the midtone. This is because the pixels causing the unwanted pattern become a much smaller fraction of the collective dot. But then, beyond the midpoint as the dot approaches the shadows, the new pixels again become a larger fraction of the shrinking hole, and the patterns become more visible.

Figure 6.17 shows a comparison of a multi-center quad dot with a simple dot. The figure represents a gray diagonal wedge going from black in the lower left corner to white in the upper right corner. The region above the diagonal has been halftoned with a simple dot, and the region below the diagonal with a four-centered dot. The simple dot shows very visible contours between levels, because it only produces 19 levels. The quad dot does

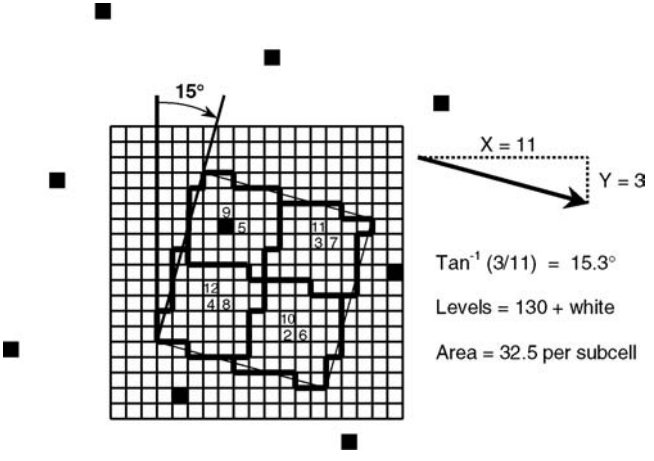


Figure 6.16 Multi-center screens.

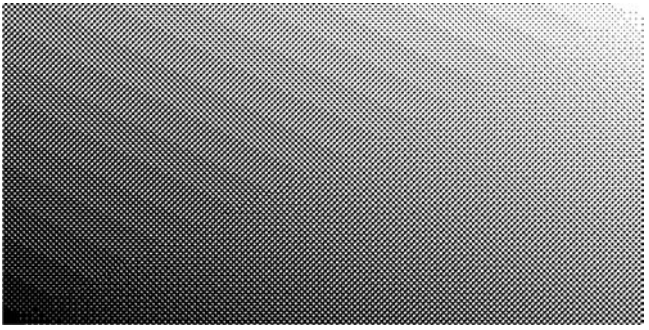


Figure 6.17 Comparison of single- and multi-center dots.

not show contours, because it produces 73 levels on the average. On the other hand, it does show some visual artifacts in the highlights and the shadows. The two dots have identical frequencies and, for every fourth level of the quad dot, it exactly matches up with the simple screen.

In general, the use of multi-center screens is a good trade-off in achieving better image quality for a given printer. The printer can use the same hardware and software and simply utilize a larger array.

6.4.8 *High addressability*

High addressability in a digital printer is where the addressability of the writing device is finer than the writing spot. It is normally used in only one direction of a raster scan — the direction that is easier to change and also the higher frequency. High addressability is essentially anamorphic resolution for the printer. High addressability in both directions would simply be called overscanning, if the writing spot were large for the scan pitch. High addressability is related to the base resolution by what is called the *addressability factor.* The addressability factor is usually an integer multiple of the base resolution such as 2:1, 4:1, or 8:1. For example, a polygon raster system of 600 lpi with a fast-scan frequency of 4800 pixels per inch would be called a 600-lpi printer with 8:1 addressability. Note that the scan line frequency is determined by the polygon and optics of the system, while the higher fast-scan frequency is achieved simply with an increase in the clock frequency along a scan line.

In Figure 6.18, you can see that there are two benefits from the use of high addressability: more gray levels and better dot shape. The figure shows a typical highlight dot using 4:1 addressability. It also shows that the writing spot has not changed from the simpler system. The dot on the right has 84 pixels instead of 21, so the dot can produce 4 times as many gray levels. And with the additional pixels, the shape of the dot can more easily approximate a circle.

Note that high addressability should not be confused with higher resolution, as the writing spot and optics of the system are usually not

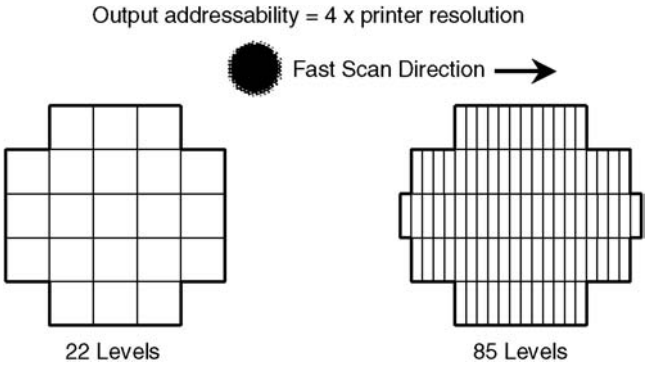


Figure 6.18 Halftone spot with high addressability.

changed. Figure 6.19 shows an idealized diagram of the performance of a writing spot as it sweeps along a scan line. When the spot is turned on and off, there is some finite rise time and fall time for the modulator response, but this time is a small fraction of the pixel time, so there is good correlation with the logical pixel stream of data. In the high-addressability case, the response times may be larger with respect to the smaller logical pixel stream. But it can be seen that, once the writing spot has been turned on, the spot can be turned off accurately in time increments of one eighth the size. Dot growth along a scan line will have eight times as many levels that are equally spaced. There is, however, a downside in the highlights where the dot has isolated high addressability pixels and also whenever

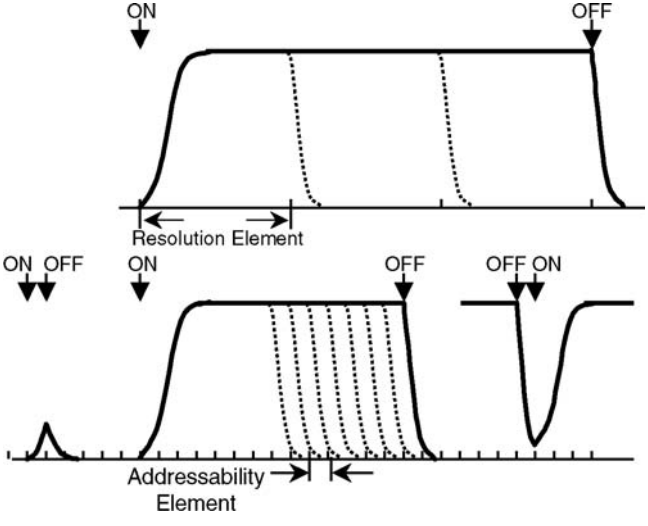


Figure 6.19 Addressability vs. resolution.

a halftone dot must grow onto a new scan line. Here, the attempt to turn the relatively large writing spot on and off too quickly may result in little or no response if the exposure does not get above threshold. There is another problem in the extreme shadows, where attempting to turn the writing spot off and back on again may result in a larger than expected off pixel. However, the dot gain will probably cause these extreme shadow levels to be lost, anyway.

The net result of using high addressability is a good trade-off — there is a significant increase of available gray levels with no impact on system optics or writing spot technology. There is, of course, an increased bandwidth requirement and a requirement for a faster output clock. The resulting printer performance curve of output density vs. requested level will have many regions of reasonably linear increase separated by flatter, less stable sections that are more susceptible to printer drift. It may be more important to calibrate such a system more frequently.

Figure 6.20 shows an example of a screen that was designed with both 8:1 addressability as well as quad multiple centers. Here, the multi-centers are used to achieve a more accurate angle, so the centers are not congruent. Notice that half of the dot shapes face down, and the others face up. This design required extensive hand tuning, because the rows of alternate up and down orientations have a tendency to appear closer together or farther apart. When this happens, the rows that are closer together form a visual unit, and a pattern at half the desired frequency appears. Notice that special effort was made to achieve a symmetric 50%-point checkerboard. The 50% black dot is exactly the same shape as the 50% white hole.

6.4.9 *Halftone grid relationships*

It is easy to be confused about the various grids that are mentioned in a discussion of digital halftoning. There are at least three grids: the basic printer raster, the image sample raster, and the halftone screen grid. One could also consider the halftone implementation grid that is a subset of the

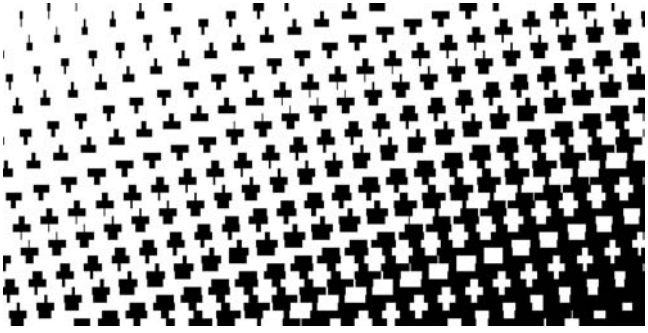


Figure 6.20 Quad dot with 8× addressability.

basic printer grid. Figure 6.21 shows graphically the relationships between these various grids.

The basic printer grid is the binary logical pixel raster that is sent to the printer after halftoning. The vertical component corresponds to the width of the scan lines of the printer. For an LED-bar printer, one should rotate the diagram 90°, as this corresponds to the width of an individual LED element. The horizontal component corresponds to the clocking interval along a scan line or the clocking interval for the LED elements. The scan line dimension must be a multiple of the clock interval dimension or there will be complications in the rational-tangent screen design.

The sample grid is the contone pixel raster before halftoning. The vertical component must be 1:1 or a multiple of scan lines. The horizontal component is usually the same as the vertical component; thus, it will be a multiple of high-addressability pixels. The pixel raster for a contone pictorial image is normally square and may be lower resolution. Images should be scanned at two or three times the frequency of the halftone screen that is used. This will ensure at least four samples for each halftone cell so as to create partial dots. The image raster may be any arbitrary resolution, but it must be resampled or interpolated up to the sample grid before halftoning.

The halftone grid is the visual dot pattern produced by the halftoning operation. It will be rotated for color halftone printing. It is made up of elements of the basic printer grid, but there is no inherent synchronization with the printer grid or the sample grid. The halftone implementation grid is made up of rectangles with the same area as the halftone grid, but it is not rotated. It is important to remember that the entire halftoning operation is essentially rectilinear horizontally and vertically, and the visually rotated screen patterns are entirely a consequence of the threshold values used in the rectangular halftone threshold array.

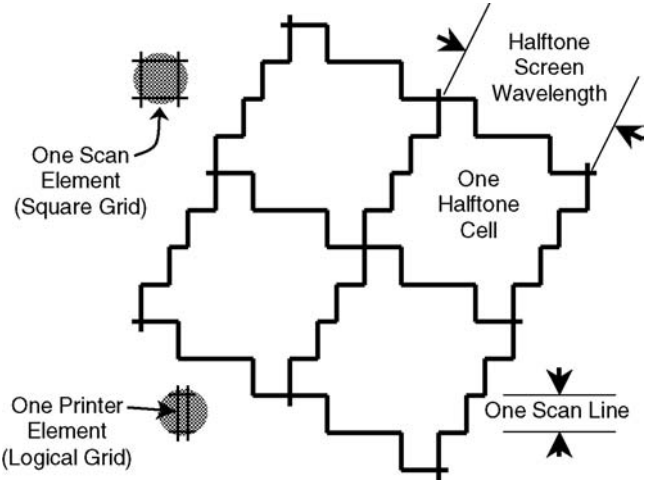


Figure 6.21 Digital halftone grid relationships.

6.4.10 Practical design considerations

This section contains some "tricks of the trade" and practical considerations for clustered halftone dot design. Some of these topics are not intuitively obvious until you see them explained. Most are common sense that come from experience in designing dots.

6.4.10.1 Periphery is the noise source

The periphery of the halftone dot is the source of most of the image noise in digital halftoning. The white paper and the solid film of colorant within a dot are seldom a problem. The irregularity of the perimeter from dot to dot is the problem. The best way to reduce noise is to design for minimum dot edge-to-area ratio. This is achieved with circular spots in the highlights and circular holes in the shadows. Single pixel appendages on dots are a particular problem.

Stochastic screens appear to violate this design principle. Stochastic screens generally have a high dot edge-to-area ratio and do indeed show a greater amount of visual noise. This unpredictable dot periphery adds to the inability to proof stochastic screens and also to the difficulty of calibrating them.

There are a number of other image defects, sometimes also called "noise," that are associated with halftone printing. They are listed here in order of decreasing visual frequency.

- Dot noise — very fine noise due to irregular edges of individual halftone dots
- Granularity — dot-to-dot differences due to writing variations and instabilities; more severe with multi-center dots
- Moiré (rosette) — the basic regular interaction of rotated color screens; affected by rational-tangent screen angles and screen frequencies
- Mottle — variations over multiple dots; may not be due to the dot design; can be caused by paper fibers or texture
- Streaks — vertical process defect due to dirty optics, charging wires, irregular LED elements, etc.
- Banding — horizontal process defect due to paper motion variation, scanning variation, etc.
- Moiré (color shift) — low-frequency interaction of halftone screens due to small variations in screen frequency and angle, two-color moirés, second-order or three-color moiré

6.4.10.2 Dot-center migration

Halftone dots should be designed so that the dot enlarges uniformly as it grows and maintains a stationary *center of gravity.* Dots that do not grow uniformly will have a stationary edge that can cause problems in image quality. A stationary edge does not produce partial dots, so image sharpness

can vary across an image. Figure 6.22 shows three examples of dots with and without dot-center migration.

The migrating dot growth shown in Figure 6.22A begins in the upper left corner of the dot and works its way down to the lower right. The migrating center of gravity is shown on the right. This type of dot growth can lower image quality. The dot growth shown in Figure 6.22B is called *spiral dot growth.* It produces a relatively low-noise dot, because it has a minimum of appendages and a compact perimeter. But it is slightly lopsided and less symmetric. The dot growth shown in Figure 6.22C is called *double spiral dot growth.* It is more symmetric but produces more visual noise due to the extra appendages. Small dots may need the low-noise characteristic more than the symmetry. Larger dots can instead concentrate on the symmetry, as the noise due to the periphery is less of an effect.

6.4.10.3 Dot cell boundary

It is not advisable to draw boundaries around a single halftone dot during the dot design phase. It is important to remember that the halftoning function is actually continuous in x and y, and any boundary is arbitrary and may mislead the designer. In particular, a square boundary can be detrimental, although it is tempting in the case of a zero-degree dot. Figure 6.23 shows a square boundary drawn around a single cell in a repeating screen pattern along with a three-dimensional view of the same cell's growth structure.

The design of the highlight portion of the dot is no problem here; the spots are well separated from each other. The problem arises after the 50% point where the dots touch. If a square boundary were drawn around the dot, then further growth of the dot must take place in four disjoint corners of the dot simultaneously. In actuality, this growth is around a single "hole" and should be considered all together.

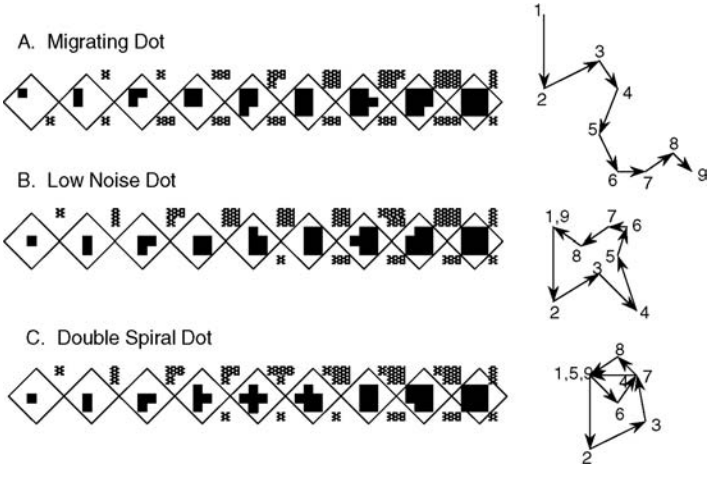


Figure 6.22 Dot-center migration.

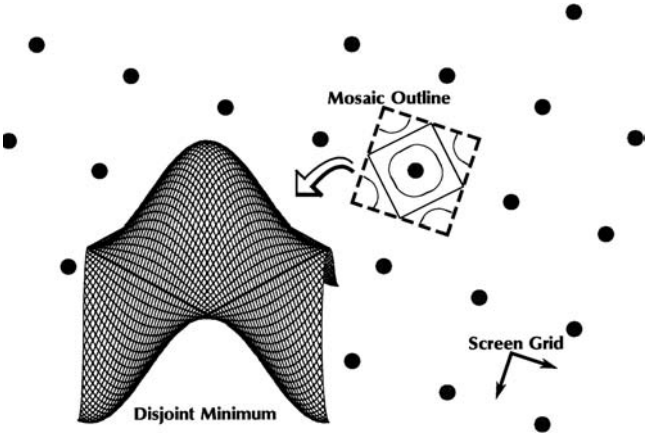


Figure 6.23 Mythical dot boundary.

Figure 6.24, on the other hand, shows the boundary drawn around the 50% boundary of the two phases of the dot and a three-dimensional view of the cell's growth structure. The best visual boundary for a halftone dot is the 50% outline, a rectangle that encompasses one highlight peak and one shadow depression. This boundary is a natural one in that the dot growth from each direction converges at this rectangle.

There is a natural phase shift in halftone dots. The growing spots are offset from the shrinking holes by this phase shift, as can be seen in Figure 6.9 in Section 6.4.4.

The only other boundary that should be drawn around the halftone dot is the rectangular tile with which it is implemented. This boundary should never be seen in the dot that is produced. Much of the hand-tuning process

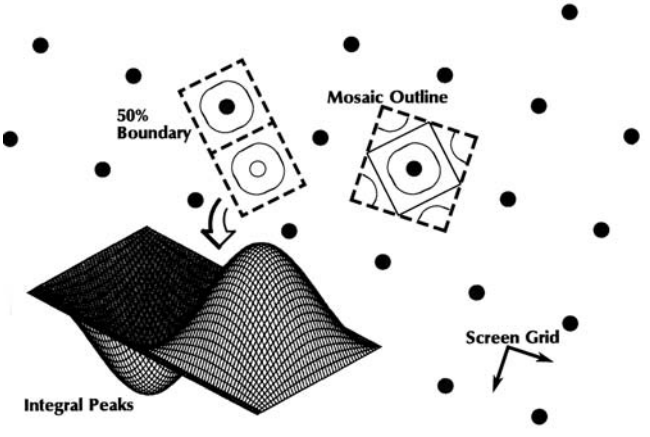


Figure 6.24 Significant 50% dot boundary.

is to remove any traces or hints of where the implementation boundary might be.

6.4.10.4 *Dot growth considerations*

The dot growth sequence of digital halftones has a strong impact on the output quality, particularly in the case of laser raster devices. Digital halftone dots at intermediate resolutions, 300 to 600 lpi, are usually carefully designed and tested to avoid visual artifacts that can easily occur. A separate sequence is designed for each dot size and angle of rotation. While a computer program can make the first iteration on dot design, improvement on the design can always be achieved after testing and measurement. Automatic rotation of halftone dots is expected to produce dots that would be inferior to carefully hand-tuned dots.

Dot growth is important for several reasons. One consideration is geometric. A digital halftone dot cannot grow uniformly as an optically variable screen can, due to the basic spatial quantization of the raster matrix. The dot can grow only by adding at least one printer-resolution pixel at a time. In doing so, it is easy for the dot to become lopsided or asymmetrical, producing visual artifacts that show up as undesirable textures or coarse patterns in the print. At higher printer resolutions, the problem is greatly reduced, because each printer pixel is an insignificant fraction of the resulting clustered dots.

Figure 6.25 shows the 50%-point for a digital 18° rotated dot. The desired checkerboard square is shown in the upper right of the figure. It can be seen that approximation of this midpoint in the dot is rather coarse, but it is as symmetric as possible, and the shape of the 50% spot is identical to the shape of the 50% white hole.

6.4.10.5 *Dot gain*

Another consideration is the nonsymmetrical nature of most raster output devices including laser raster systems. Figure 6.26 shows that a laser beam

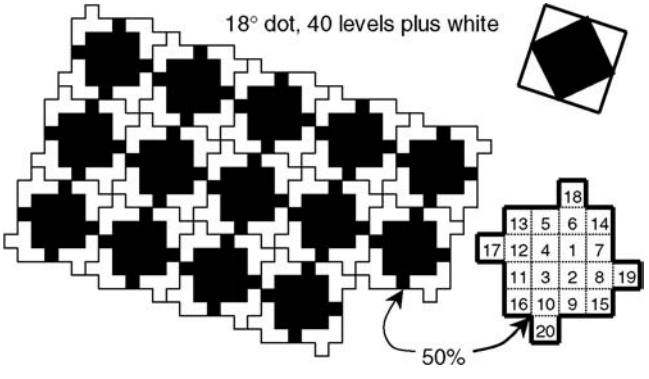


Figure 6.25 Mid-point checkerboard.

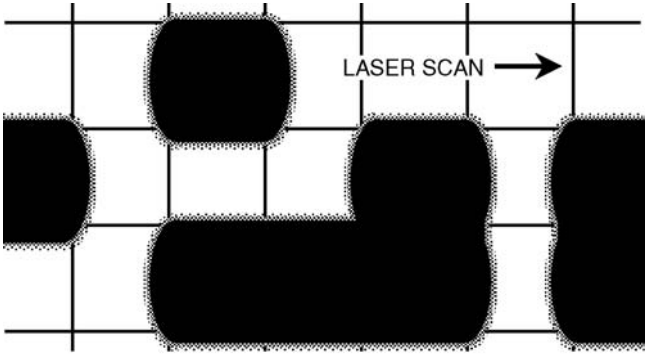


Figure 6.26 Dot gain from laser scanning.

exposes a photoreceptor by sweeping across one scan line at a time. The laser beam is Gaussian in exposure profile and larger than the width of the scan line so as to fully expose between adjacent lines. The beam is also nearly circular so, when it is turned on for a logical pixel, it may also partially expose the previous pixel. And when it is turned off, it may have already partially exposed the next pixel. The net result is that the halftone dots that are produced are slightly larger than the logical data sent to the modulator.

Laser beam shape and profile, modulator response characteristics, and scan line overlap all affect the symmetry of digital halftone dot growth. It has been found that two printer pixels together along a scan line can have a much different contribution to print density as compared with two printer bits together on adjacent scan lines. Hence, dots that are not carefully designed can have sudden jumps in print density as well as many levels with little or no density change at all. While most dot gain is positive growth, high-addressability writing may produce some negative dot gain in isolated pixels.

There are a number of sources for dot gain in electronic printing systems, as follows:

- Laser beam size and shape
- Addressability not commensurate with beam size and raster pitch
- Photoreceptor voltage spread
- Development of toner or inking
- Intermediate transfer on some devices
- Transfer to paper
- The fusing, fixing, or drying process

Not all dots grow to the same degree. The largest dot gain is in the midtones (40 to 60%), where the perimeter of the dots are at a maximum. The perimeters

of the dots are a less significant fraction of the image in both the highlights and in the shadows.

Dot gain is also more pronounced at higher frequencies, where the physical size of the dot spread is larger relative to the dots being produced. This is why dot gain may be a big consideration in screen frequency choice.

There is also an optical effect that occurs in printing on paper. After the dot is printed on the paper, the eye perceives this printed dot as larger than actually printed. This effect is called *optical dot gain* and is a result of internal scattering within the paper. The light is scattered within the paper, and some of the light is absorbed on the underside of the toner or ink. This phenomenon is also referred to as the Yule–Nielsen effect,[18] and it has the same effect as printing a larger dot. Optical dot gain depends on the opacity and the surface of the printed substrate.

Fortunately, for clustered dot printing, the dot gain is predictable and can be compensated by the calibration process. Hand tuning the dot design during testing can minimize irregular dot growth.

6.4.10.6 Dot calibration considerations

The net result of irregular dot growth in a digital raster printer can be summarized in the simple statement, "Not all levels are equal." This nonequal nature is further affected by the normal dot gain characteristics of the rest of the printing process. This is true of both lithography and xerography as well as inkjet printing. It is difficult to predict exactly how a particular digital halftone image will print on any random printer. For this reason, it is important that the characteristics of the printer be known prior to the actual creation of the halftone bit-map. The characteristics can be determined by processing, printing, measuring, and analyzing test targets with the same film, plate, drum, inks, and/or toners that are to be used for the desired print.

The use of a printer-independent file format is especially important from this standpoint. It is known that a halftone bitmap created specifically for a high-resolution film printer cannot be subsampled and printed adequately on a lower resolution device. Even the line graphics and fonts suffer greatly in a down-conversion process, where a complicated conversion algorithm such as "black always wins" may be necessary to ensure that fine lines are not lost altogether. Going from the printer-independent file to a bitmap for each device will always produce higher quality. Dot calibration is further discussed in Section 6.4.13.

6.4.10.7 Dot frequency

From a psychophysical standpoint, it is known that the need for image resolution is image dependent. Pictorial images and images with large flat color areas require grayscale resolution to avoid contours. Text, line art, and images with much edge detail require spatial resolution, grayscale is less important. Image type should be considered when choosing the appropriate halftone frequency. High-frequency halftones have fewer grayscale levels

but can more closely reproduce detail. Low-frequency halftones have more grayscale levels but add a visual noise to the image.

6.4.10.8 Dot levels

The human visual system can detect at least 64 grayscale levels. Some people wrongly assume that this means that 64 grayscale levels are adequate for a halftone dot. The actual capability has to do with "just noticeable differences." While just noticeable differences are considered to be equally spaced in visual space (i.e., CIE L^*), they are not well spaced in either reflectivity or density space. Digital halftone dots are most closely linked with reflectivity space, as the area coverage of the dot produces the grayscale. Unfortunately, a one-pixel step in area coverage for the highlight region is considerably more visible than the same one-pixel change in the shadow region. In other words, the dots are not equally spaced visually. Because of this, many more than 64 grayscale levels will be required to achieve smooth gradations without contours. Dots with about 128 levels are practical but can still produce contouring in the highlights. The image-processing path normally carries only 8-bit data or 256 levels, but it is advantageous to create dots with as many levels as possible so that the 256 data levels can be mapped into the most appropriate halftone levels.

6.4.10.9 Screen angle

The human visual system is least sensitive to a screen pattern angled at 45° and most sensitive to one at 0°. The 0° pattern should simply never be used, except when required in a rotated dot set for color, and then only in the yellow separation where it is not visible. The 45° screen is just as easy to implement as the 0° screen and is much more effective.

6.4.10.10 Visual trade-offs

The biggest halftone trade-off is between grayscale levels versus frequency. Again, you must balance the requirements for numbers of levels with the requirement for edge detail and sharpness. This trade-off may be different for each printing system considered, especially as the basic resolution is changed. The use of multi-centered dots to increase the grayscale levels is another trade-off, as increasing the number of dot centers adds lower-frequency visual artifacts.

Another visual trade-off is the size of the rosette when printing color with rotated screens. Sometimes a suboptimal individual screen for one separation may result in an overall improvement in image quality if it reduces the visibility of the rosette.

6.4.10.11 Dot shape

The most successful halftone dots have been circular in the highlights and in the shadows. Peter Stucki (IBM Zurich Research) reported, "Experiments have shown that the visibility and the perception of different shape dominant

patterns may be subjectively disturbing in bilevel rendition of continuous-tone data." Circular dots have the smallest perimeter-to-area ratio and hence the smallest chance for error and visual noise. The only symmetrical transition from circular dots to circular holes is the familiar checkerboard pattern at the midtone. The checkerboard is sometimes distorted by using "elliptical" dots when the density jump at the midpoint is too noticeable. However, elliptical dots produce a "chaining" effect that can be more visually noticeable. The unpredictable dot shapes produced by stochastic screening are another reason to restrict their use to the special cases when they are required.

A three-dimensional representation of the classic halftoning function is shown in Figure 6.27. Several cycles of the function are shown, partly to stress that it is a continuous function that repeats across the page. There is no reason to associate one "hill" with either "valley." The key thing is to supply the appropriate threshold value to compare with the image data at each printer pixel. It does not matter how the threshold was determined or what neighboring thresholds might be.

It is not surprising that this function appears very sinusoidal. It can be described as a function of two cosines in the simplest zero-degree case as follows:

$$K = \cos 2\,\pi\, f_x\, x + \cos 2\,\pi\, f_y\, y$$

Figure 6.28 shows a topographical plot of a single zero-degree cell. It shows a circular shape in the center highlight region, progressing to the square checkerboard at the 50% point, and then progressing to a circular shape in the shadow region. The shadow "valley" is broken into four parts here and distributed into the four corners for this cell. When this cell is tiled together to cover the page, the four corners from adjoining cells form the valleys with no visible seams.

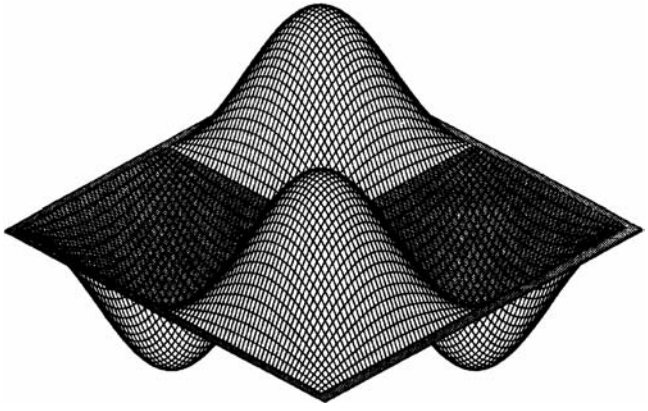


Figure 6.27 Halftone threshold function.

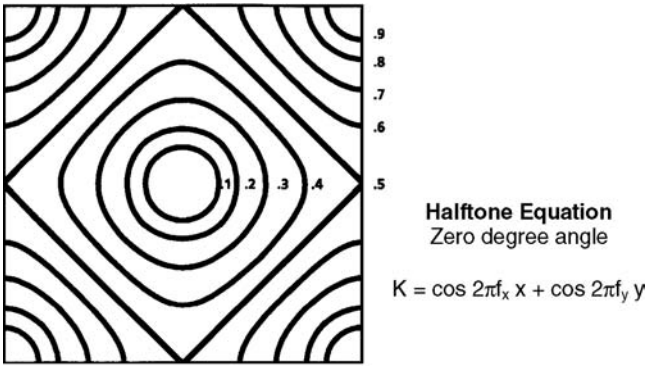


Figure 6.28 Halftone function equation.

The functions shown here are the theoretical ideal in that they are shown as very smoothly varying functions. In a digital representation, they might not be so smooth, because they are made up of a finite number of pixels that can only approximate the desired ideal shape.

6.4.10.12 Where the dots touch

In particular, the 50% point where the dots first touch each other is a difficult design problem. Figure 6.29 shows a simple 45° dot growing to the 50% point. The 45° dot is the easiest to design here, as the checkerboard square is aligned with the output raster, but the 50% point has other problems as well.

The plot on the left in 6.29 shows that, even in lithography, where the dots are not digital, there is a characteristic change in slope for the printer response near 50%. This anomaly, predicted by Hunt[19] before digital printing began, may be partly due to an increase in optical dot gain as the dots change from growing spots to shrinking holes.

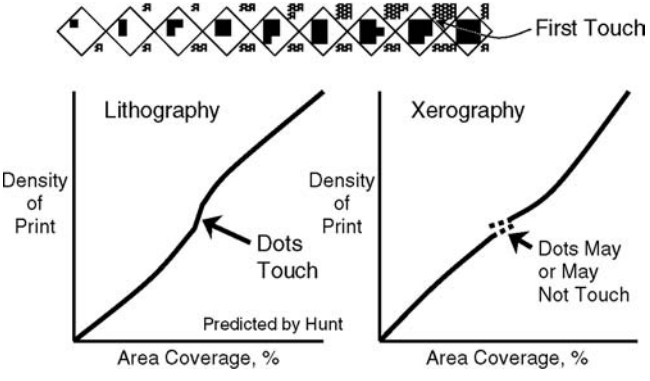


Figure 6.29 Where the dots touch.

The plot on the right side shows that a noisy digital system like xerography not only has the predicted change of slope but also a pronounced increase of noise. This is partly due to the sharp edges of the 50% dot with its high-frequency content. But it is also due to the unpredictable nature of xerography, wherein the dots may either be drawn together or appear to repel each other, depending on local conditions. These unpredictable conditions include charges on the photoreceptor, transfers from photoreceptors to the print, possible intermediate transfers, paper fibers on the print media, and fuser behavior.

There is also the difficult design goal of keeping not only the spots symmetrical but also the increasing need to keep the white holes symmetrical as well.

There are several strategies for dealing with the dots touching. Figure 6.30 shows two different strategies. Notice that the two dot-growth methods shown in the figure result in exactly the same angle and frequency and the same number of levels. But they demonstrate different characteristics where they touch.

On the left, the first connection between dots in the dot-growth sequence is a solid connection — a pixel that bridges between dots on two sides. This is a very low-noise connection. The dots are cleanly separated without it and clearly connected with it turned on. The possible problem with this strategy is that the connection is too good and that it may exacerbate the increase in dot gain that is also happening at the 50% point. Because of this possible jump in printer response at this level, as shown in the inset plot, this strategy might be called *boom dots.*

On the right, the first connection between dots is a noisy connection — a pixel that is well connected to one spot but just barely touches the other spot on the corner of the pixel. This configuration increases the noise potential because of the problematic nature of this unpredictable connection. On the other hand, the benefit of this strategy is that the noisy connection may

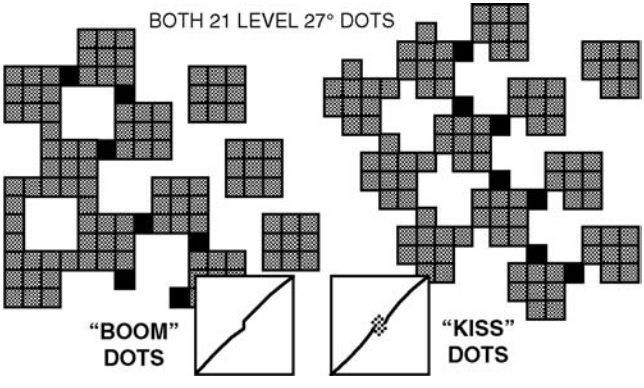


Figure 6.30 Strategy at the mid-point.

mask or hide the sudden jump in dot gain. Because of the barely touching nature of this method, this strategy might be called *kiss dots.*

These are both viable strategies and might be used in different cases. For high-frequency screens with fewer levels, the kiss dots may be more appropriate, because the jump in response would be significant fraction of the total range. The increase in noise might be preferable to a visible contour in the image. For low-frequency screens with many levels, the boom dots may be preferred. With many levels, the jump in response would be insignificant, and the reduced noise would be a definite benefit.

Another way to reduce the problem at the 50% point is the use of elliptical dots. An elliptical dot does not have an exact checkerboard at the mid-point, but dots touch in one direction first and then the other. This can result in changing one significant tone contour in a troublesome area into two smaller contours in different areas. The downside of elliptical screens is the appearance of chaining in one direction when the dots begin to touch.

6.4.10.13 Data precision and file size

Psychophysicists have determined that the human eye can perceive at least 64 shades of gray. This fact has often been misinterpreted as a requirement for data precision. However, the number refers to shades of gray that are equally distributed in human visual space, a condition not met by either the scanner or the printer. CCD scanners scan in reflectivity space. This can be observed by noticing that the raw scans are generally very dark, with most of the data lumped in the shadow region. To be useful, the scans must be transformed with either a logarithmic function (to convert to density) or a cube root function (to convert to L^*). If the scanner delivered data with 8 bits per pixel (256 shades) in each of RGB, the data become dangerously close to human detectable quantizations in some areas of the range after transformation. But, in addition, further processing for a tone reproduction curve (TRC), image aesthetics, color correction, and printer dot gain calibration all result in further quantizations of the data. With all these considerations, it is clear that eight bits per pixel from a CCD scanner is not enough.

State-of-the-art scanners now available allow a downloadable table look-up that permits the data to be transformed from reflectivity into something closer to human visual space before the data leave the scanner. For example, the scanner may scan with 10 to 14 bits of precision in reflectivity for each of RGB but transform to 8 bits of density delivered to the computer. In this case, with well-conditioned data, eight bits of RGB should be adequate for desktop publishing but not enough for high-end work.

If the data are converted to colorimetric form, they are the most compact and also printer independent. Images can be efficiently represented in 24 bits of $L^*a^*b^*$. If they are converted to CMYK, they require 32 bits of dot percentage, and they are printer specific. XYZ data, while colorimetric, is not linear in a human-visual sense and requires more bits per pixel to avoid quantization during further processing.

Grayscale and resolution requirements are a function of image type. Pictorial color (natural images) has high grayscale but low spatial resolution requirements, while line art and text have the reverse.

The eight bits per pixel in each color mentioned above are necessary for pictorial color, but the number of pixels per inch can be very low. A rule of thumb in the graphics arts industry has been that scanned natural images require only twice the spatial resolution of the halftone screen that will be used for printing. If the printing process utilizes a 150-dpi screen, then scanning images at 300 pixels per inch should be adequate. Desktop printers with coarser screens can use even lower resolution for images. This method ensures four input samples for each halftone dot and allows "partial dots" to form, which can reproduce higher frequencies than that of the halftone screen.

On the other hand, spatial resolution requirements for text and line graphics are much higher, while their grayscale requirements are lower. Good typography requires at least 800 pixels per inch to avoid detection of "jaggies" by the human eye. This is due to the fact that, in high-contrast areas of a scene, the eye is most sensitive to edge detection and linear acuity but less sensitive to color and grayscale. Text and line art can be adequately represented with only one or two bits per pixel. Rarely are high spatial resolution and high color precision both needed at the same time. Some block-truncation compression schemes take advantage of these characteristics.

Some grayscale in scanned text and graphics is useful in that it softens the pixel edges and prevents aliasing and jaggies from forming. But color fidelity is less of a major consideration in high-contrast areas of a print.

6.4.11 Angle family

Most of the halftone technology that has evolved for conventional printing is still valid in the printing world, but there are some differences. While contact halftone screens can be placed at any angle and have any arbitrary frequency and high grayscale capability, digital halftone screens are normally an engineering trade-off between spatial frequency and number of grayscale levels that can be produced. It is also very difficult to achieve some screen angles — in particular, the desirable ±15° screen angles that are used in conventional printing. Much of the recent technological energy has gone into solving this particular problem, with mixed results.

Digital printing systems still have problems with process color registration, although some are better than conventional printing presses. Unless registration is perfect, there will be a problem in color fidelity, both from area to area on the same print and from print to print, unless rotated screens are utilized. Some think that "perfect" is a relative term but, in fact, as screens approach perfect alignment, color moiré and screen moiré can become increasingly visible.[20] There has been a recent revival of frequency-modulated screening techniques (stochastic screens) that may reduce or eliminate moiré problems. However, these methods rely on uniformity and stability

of the smallest printing unit or pixel at the finest resolution. Dot gain, paper fibers, toner size, and laser beam uniformity all have taxing effects on the finest printer dot. Conventional clustered halftone dots limit these effects to the extreme highlight and shadow regions, while FM screening has problems throughout the grayscale range. While the initial impression is one of sharpness, color fidelity and the ability to calibrate may be problems.

6.4.12 Moiré considerations

Printing with rotated halftone screens often produces moiré in color prints due to the interaction of the screen angles and frequencies. Moiré can be exacerbated in digital halftoning by the approximate screen angles and frequencies that are used by the rational-tangent screen algorithm.

6.4.12.1 Two-color moiré

The most common moiré is due to the use of the zero-degree screen for the yellow colorant. When used with the usual 15, 45, and 75° screens for cyan, black, and magenta, the yellow is only 15° away from the cyan and magenta angles. The two-color moirés then appear most commonly in light greens and pinks and tans. One method of avoiding this moiré is to change the frequency of the yellow screen. See Section 6.6 for further discussion of moiré.

6.4.12.2 Three-color and four-color moiré

Other than the two-color interactions, more complicated moiré patterns can develop due to the use of rational-tangent screens that only approximate the desired angles. Finding a compatible family of screens for a given printer is often a difficult problem. Much experimentation is required and often will lead to suboptimal screen frequencies but lower moiré.

6.4.12.3 Auto-moiré

Auto-moiré is the condition in which the screen beats against itself. This can most easily occur if the screen is designed by a computer program. The best solution is to hand tune the design.

Another very common form of auto-moiré is the beating of two screens derived from the same design. It is common practice to design a +15° screen for perhaps the cyan separation and to then flip the angle for use with the magenta at 75°. Unfortunately, then every anomaly of the first screen will also occur for the second screen and on the same scan line. It would be better to design the second screen from scratch and vary the design. Changing the phase of the second screen can also help to relieve the problem.

6.4.13 Calibration

The use of digital halftone screens requires them to be calibrated for the printer with which they will be used. This is particularly important when

the printer has a considerable dot gain characteristic or when the screens have only a minimal number of grayscale steps.

Strictly speaking, calibration is an ongoing process that should compensate for the current printing conditions. Instead, most desktop software sold for printer calibration is actually based on printer characterization and is only a one-time setup. The differences between calibration and characterization can be summarized as shown in Table 6.3

Calibration is a short-term process that should be performed more often as the printer drifts in performance. Characterization is a much more stable process and only needs to be done once. Characterization is most important for color halftoning. The attributes of calibration vs. characterization are as shown in Table 6.3.

Table 6.3 Characterization vs. Calibration

	Characterization	Calibration
Stability	Stable with time (assumption)	Short-term drifts and environmental sensitivity
Process	Time consuming	Real-time, repeatable
Sensors	Expensive colorimetry	Inexpensive densitometry
Complexity	Three-dimensional or four-dimensional problem [3×3 matrix, 3-D lookup table (LUT) with interpolation, includes black]	One-dimensional problem (four LUTs)
Required by	Colorant characteristics, halftone orientation strategy	Dot gain, electrical and mechanical drift, d_{max}
Detail	Smooth functions	Detailed functions (can contain kinks and flat spots)
Method	Statistical averaging process	Measurement process

6.4.13.1 Screen threshold assignments

The first step in halftone screen calibration is to assign threshold values to each level that can be produced. It is possible to include the actual screen linearization calibration in these thresholds, but the calibration is usually done separately by means of a tone reproduction curve (TRC). The TRC is a table look-up. The TRC allows the calibration procedure to be separated from the halftone dot design and to be performed more easily and frequently.

When the image path is using 8-bit precision and the halftone can only produce 256 gray levels or less, then it is convenient to simply space the thresholds equally between 0 and 255. This is equivalent to spacing them in area-coverage space. In this case the spacing is unimportant, because the TRC can address any of the gray levels, and the calibration will make the appropriate mapping.

When the halftone can produce more gray levels than the precision of the image path, the threshold assignment becomes important. This is because, with n bits, only 2^n threshold values can be used, so some of the halftone's gray levels will not be addressable and cannot be used. In this case, a nonlinear spacing is desirable. It is best to utilize all of the highlight gray levels when possible and to sacrifice some of the shadow levels if necessary. Missing levels in the highlights are much more apparent than in the shadows.

6.4.13.2 Individual screen calibration

You can think of tone reproduction and gray balance as representing the central axis of color space. Getting these correct can do more to make reproductions look right than any other aspect of the subject of color reproduction.

There are a number of sources for dot gain in electronic printing systems, as follows:

- Laser beam size and shape
- Addressability not commensurate with beam size and raster pitch
- Photoreceptor voltage spread
- Development of toner or inking of plates
- Transfer to paper

Note that dot gain can be negative as well as positive; i.e., the dot can be reduced from the desired size. While each source can contribute to overall dot gain, the effect is corrected with a single TRC that maps desired grayscale into available grayscale dot levels. The TRC can be a look-up table (LUT) through which the desired grayscale is passed. But, alternatively, the TRC can be combined with the halftone threshold array so as to effectively *linearize* the digital dot. LUTs can then be used for aesthetic correction or gray balance instead of the detailed compensation of the halftone levels. This can be important, as the optimal correction for a digital screen requires fine detail in the TRC curve and intimate knowledge of the printer characteristics. In addition, some color correction schemes depend on the individual screens to be linear in density.

For a black-and-white printer, or for a color printer that relies on individual separation linearity, calibration simply means distributing the halftone levels linearly with respect to some measurement space. The levels are already distributed in area-coverage space, but this is of little value because of dot gain and printer drift. The printing industry has long used density space for linearization. It is becoming more popular to use a human-visual space such as CIE L^* to allow the gray levels to be used most efficiently.

Figure 6.31 shows the procedure for single-separation linearization. Prints are made with a series of test patches that span the range from white to solid colorant. The test patches are measured and plotted, shown in the left plot as L^*. These data are then normalized by stretching the white point and the darkest point to full scale. In the case of L^* measurements, they are

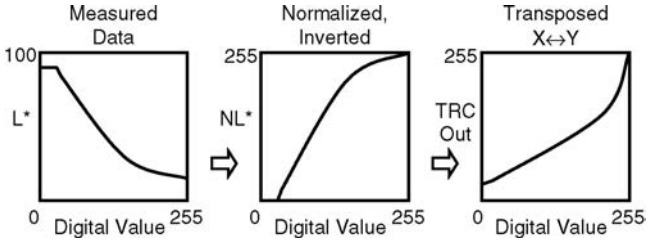


Figure 6.31 Single-separation linearization in CIE L^*.

inverted so that white is at the origin. The normalized and inverted data are shown in the middle. The data are then simply transposed, so the x axis becomes the y axis and vice versa. This becomes the linearizing TRC shown on the right. If patch values are passed through this TRC and then reprinted, the resulting plot of the new L^* measurements will be linear.

The normalizing step may also include an "aim" curve to compress the shadows if the printer lacks density and dynamic range.

Linearizing each colorant halftone separately does not ensure that the neutral axis is straight. Figure 6.32 shows conceptually that, on the left, the cyan, magenta, and yellow vectors in the color space are calibrated, but the neutral diagonal vector can be somewhat curved through the space. This might be due to the halftones or the spectral characteristics of the colorants. This nonlinearity in the neutrals would have to be corrected in the color correction stage by means of the color characterization.

An alternate calibration method is the *neutral balance* calibration. Shown on the right, by calibrating all three colorant halftones together, the neutral diagonal becomes straight. This means that equal-valued triplets of cyan, magenta, and yellow will produce neutrals. Some color characterization methods can work more efficiently in this *equivalent neutral* space.

6.4.13.3 *Neutral calibration*

When printing, apart from calibration of the printer, it is useful to take steps to guarantee that, when equal values C = M = Y are sent to the printer, a

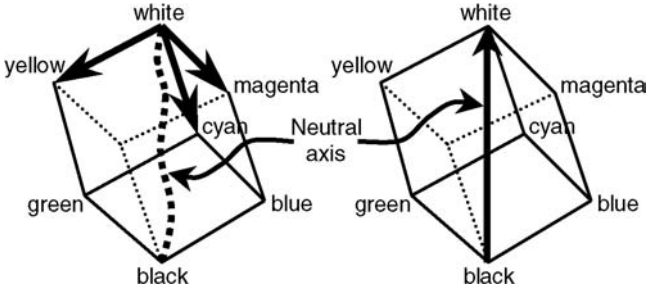


Figure 6.32 Neutral axis effects from two calibration methods.

neutral is produced on the print. An important consideration, often overlooked or downplayed, is gray balance on a printer.[21] The human eye is very sensitive to differences in color in the region near gray. Gray balance on the printer is accomplished by printing a series of near neutrals on the printer. A set of five or seven values is selected to print magenta patches in the region near a near-neutral. The same wedge in yellow is printed at right angles to the magenta. A uniform cyan patch with the average of the five or seven values is printed over the crossed wedges of magenta and yellow. It is hoped that a gray color, $a^*b^* = 0.0$, will be printed on one of the 25 to 49 patches. All areas of the test chart are measured with the colorimeter. Select the patches closest to $a^*b^* = 0.0$, and interpolate the printer driver values that would be expected to produce $a^*b^* = 0.0$ and the corresponding L^*. Repeat this procedure for additional values of L^*. Six or seven additional samples may be enough initially to plot curves of C, M, Y vs. L^* to determine the values of CMY necessary to produce neutral throughout gray scale.[22] Three one-dimensional tables are then created to guarantee that equal values for CMY produces a^*b* 0.0 on the print.

Figure 6.33 shows the CMY triplet values plotted with their resulting L^* value. As in the previous method, three smooth curves are drawn, and then the data are inverted, normalized to white and black, and transposed to produce three TRCs — one for each of C, M, and Y.

6.4.13.4 *Color characterization*

Color correction is the step that takes image data from red, green, and blue values into the cyan, magenta, yellow, and black values that are then halftoned. Color correction makes use of the color characterization data. There are several methods used for color correction. The simplest method is the use of a 3 × 3-matrix operation. More complex methods make use of three-dimensional look-up tables and interpolation.

There is an order of importance for correcting images for printing.

- The first step is *neutral balance*. Any color cast in the image should be removed before further processing. Three individual table look-ups should be able to remove a color cast in RGB space.

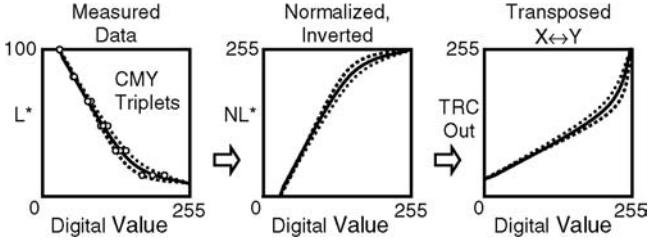


Figure 6.33 Neutral axis linearization in CIE L^*.

- The second step is *tone control*. Tone control is a luminance operation. A single lookup table is applied to the three separations in RGB space or to the L^* separation in CIE space. Tone control must correct for the scanner as well as any aesthetic tone corrections.
- An assumed step is the *linearization* of the halftone screens.
- The final step is the *color correction*. This is actually a color transformation, not a correction. This step takes the RGB or $L^*a^*b^*$ image data into the cyan, magenta, yellow, and black that are needed for printing. This is a much easier problem after the previous steps are accomplished.

6.4.13.5 *Printer models*

See Section 6.9 for a discussion on the use of printer models in halftone calibration.

6.5 *Halftone effect on color gamut*

It is common practice in the printing industry to use four-color halftone processes to produce color tonal renditions. The dot screens are angled 15 or 30° relative to one another. When only the three process colorants are used, these angles produce overlapping dots with a random distribution of eight colored areas: white, cyan, magenta, yellow, red, green, blue, and black. As a result, the average color of a given area does not change if there is a change in registration of the three colorant separations.[23]

The use of rotated halftone screen angles produces a moiré pattern called a *rosette*. Registering dot screens at 30° angles minimizes the size of the rosette, but the pattern is much more visible than the screen pattern by itself.[24,25] As the halftone frequency is increased, the moiré pattern becomes less objectionable.

Because digital high-frequency screens are difficult to produce, it is attractive to consider other halftone arrangements. Two alternatives are often proposed. One is to print the halftone dots on top of each other (dot-on-dot), and the other is to print the halftone dots with minimal overlap (dot-off-dot). Both of these methods require that the printer be able to register precisely because, in these cases, small changes in registration produce significant changes in the color of a given area.[26]

The halftone orientation technique strongly affects the color gamut and image texture of mixed color dots. A halftone model was created to compare the potential color gamuts of three halftone orientations.[14] The conclusions are also applicable to line screens and stochastic screens.

6.5.1 *Orientations*

Figure 6.34 illustrates the three different halftone orientations. In the *dot-on-dot* case, each halftone dot combines two colorants (in this case, cyan and magenta) one on top of the other, making blue. In the *rotated* version, the

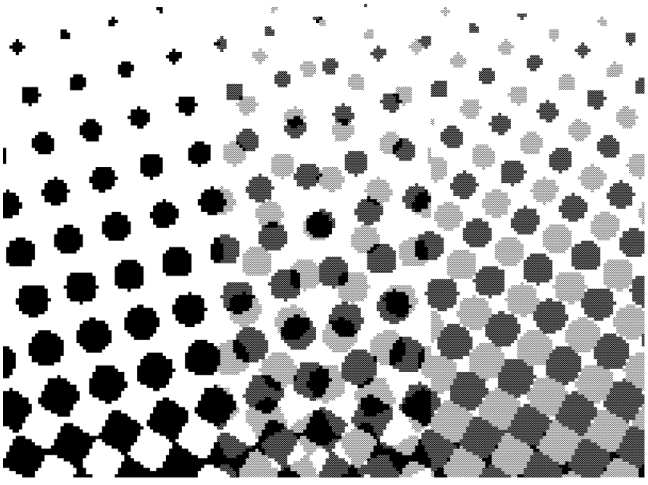


Figure 6.34 (***See color insert following page 430***) Halftone dot orientations.

screens are angled 30° relative to each other, producing a random distribution of white, cyan, magenta, and blue (cyan plus magenta) areas. In the *dot-off-dot* case, the cyan and magenta screens have the same angle, but they are positioned so that there is no overlap until each colorant occupies more than 50% of the print area.

6.5.1.1 Dot-on-dot

In dot-on-dot printing, all of the color separations use the same halftone screen and are printed one on top of each other. The preferred angle in this case would be the less noticeable 45° angle. This orientation is very sensitive to misregistration due to register error, paper speed variation, multiple raster scanner variation, and vibration. Any slight misregister forms a low-frequency pattern sometimes called *color banding.* The primary color is a strong function of the unwanted absorptions of the colorants, in their worst configuration where they have maximum interaction. The order in which the colors are laid down also affects the color because of different scattering coefficients. This orientation has the smallest color gamut, partly because of the unwanted absorptions and partly because this configuration has the most white paper visible which desaturates the colors. The color variations resulting from misregister are actually cases in which the gamut improves.

6.5.1.2 Dot-off-dot

In the dot-off-dot case, the screens have the same angle, but they are positioned so that there is minimal overlap of the colorants, especially in the highlights. This orientation is less sensitive to the laydown order, but it is

as sensitive to misregistration as the previous method. Dot-off-dot potentially has the largest color gamut. There is less interaction of the colorants and less white paper to desaturate the vibrant colors. The color variations resulting from misregister are cases in which the gamut decreases from its optimal condition.

6.5.1.3 *Rotated dots*

In the rotated dot case, the screens are angled about 30° relative to each other, producing a random distribution of the overlapped permutations of the colorants. There is some effect due to the laydown order. This orientation is very insensitive to misregistration. The gamut is somewhere between the other two orientations and is very stable. This is the most common orientation used in the printing industry where the screens are exactly the same frequency and rotated precisely to 30° separation. It is difficult to achieve the appropriate frequencies and angles in digital printing where rational-tangent angles are required, so there are generally larger rosettes and greater moiré.

6.5.1.4 *Stochastic dots*

The stochastic dot case is most similar to rotated screens in that the color overlaps are random. The color gamut should be similar but less stable due to the unpredictable dot gain. Most of the following conclusions about color gamut for rotated dots will apply also to stochastic dots.

6.5.2 *Model predictions*

The range of colors that a printer can produce is called the *color gamut* of the printer. The gamut is determined largely by the characteristics of the colorants or toners used, but it is also affected by other important factors such as gloss levels, colorant density, and the halftone dot placement scheme. Many think of the color gamut as the outline of colors in a CIE a^*b^* graph. But, in actuality, the color gamut of a printer is the envelope of the full three-dimensional color solid with dimensions of luminance and chrominance as seen in Figure 6.35. By projecting the data onto a two-dimensional plane, important information concerning color gamut is lost.

The gamut can be thought of as a cube standing on end so that the diagonal from white to black is vertical. The diagonal represents the neutral colors on the L^* axis. When viewed from above, or when projected onto the a^*b^* chromaticity plane, six of the corners are visible and form the more familiar hexagonal shape.

The model considered two-colorant combinations ranging in dot area from 0 to 100%. After the two colorants are at maximum coverage, the third colorant is then added to progress to the darkest color obtained with the colorant set.

The color gamut of a real process is considerably distorted from an ideal cube in CIE $L^*a^*b^*$ space. Yellow is much higher in lightness than either cyan

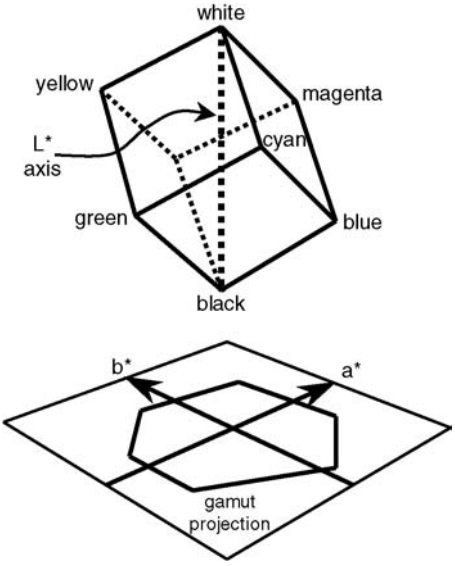


Figure 6.35 Three-dimensional gamut diagram.

or magenta, and blue is much lower in lightness than either red or green, but the edges are nearly straight. The reason for this is that, on the 12 edges of the gamut, only one colorant at a time is halftoned. For example, on the edge progressing from blue to black, both cyan and magenta are already at 100%, and only the yellow colorant is changing from 0 to 100%.

Considering the projection onto the a^*b^* plane is misleading, as only chrominance is addressed. Luminance information is not shown. In fact, the boundary of a halftoning printer's gamut is the same on the a^*b^* projection, no matter what the halftone screen orientations are, even when the three-dimensional gamuts vary dramatically. Again, this is because there are no halftone interactions between screen orientations on the edges of the color gamut.

On the faces of this hypothetical cube, the behavior is quite different. Depending on the orientation of the screens, the cube can be puffed up like a soccer ball or caved in like a used salt lick. The differences in color gamut are seen only on the faces of the cube, where the surface may be concave or convex.

In Figure 6.36, there is a large difference predicted in the chroma, c^*, between dot-on-dot and dot-on-dot configurations for the cyan–magenta colorant combination. The rotated dot case lies between the other two, approaching the dot-off-dot case in both the highlights and in the shadows. The chroma increases on the topside of the gamut up to the 100% coverage of the two colorants (blue). When the third colorant is added, the chroma begins to decrease on the underside of the gamut where the three halftone cases are equivalent from blue to black (only one colorant being halftoned).

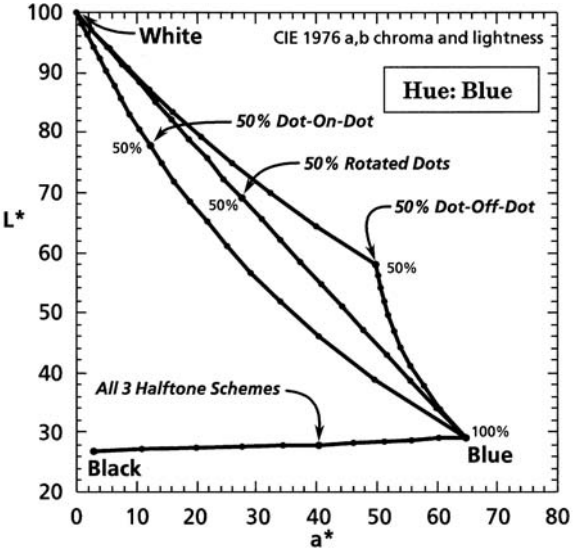


Figure 6.36 CIE L^* vs. chroma for three halftone orientations.

This plot represents a vertical slice through the three-dimensional color gamut. The space between the curves shows the increase or loss of color gamut volume due to halftone orientation. In particular, note that the 50% point for each of the curves is considerably separated in both chroma as well as lightness. The predicted sharp discontinuity at 50% coverage is a unique characteristic of the dot-off-dot configuration. It is obvious that the saturation of the dot-on-dot halftone configuration is much lower than the dot-off-dot configuration, while the rotated dot configuration is intermediate.

The a^*b^* projection in Figure 6.37 provides insight into the hue differences among the configurations. A straight line on this graph that originates at $a^*b^* = 0,0$ implies that the hue is constant. The blue hue of the dot-on-dot configuration bows toward the cyan. The hues of the other configurations are redder, particularly the dot-off-dot configuration. Note again the predicted discontinuity at 50% for the dot-off-dot.

The predicted gamut effects are similar for both the cuts through the red and the green hues of the color cube. The gamut activity is reversed top-to-bottom for the cyan, magenta, and yellow hues as, in each case, there is only one halftone on the top side of the color cube, and the interaction between screens is on the bottom faces of the cube for these three hues.

6.5.2.1 *Sensitivity to registration*

Figures 6.36 and 6.37 illustrate the changes in gamut, both in chromaticity and lightness, that would occur with changes in registration. The largest change falls on an imaginary line connecting the three 50% coverage points. These are the changes that would appear visually for identical separations

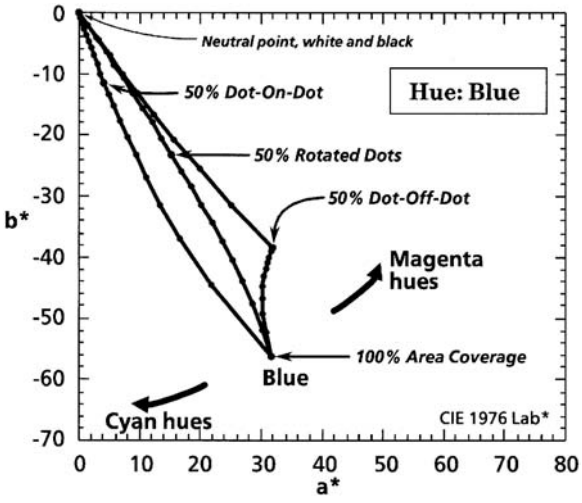


Figure 6.37 CIE a^* vs. b^* for three halftone orientations.

that are only displaced with respect to each other. With displacement of only one-half of a halftone cycle, the appearance of the dot-on-dot case would change to the dot-off-dot case and vice versa.

A good example of the possible changes can be seen in Figure 6A (see color insert). In this figure, the same photograph is printed in three halftone orientations. The examples show the change in color that can occur with change in registration for the different orientations. The photographs on the left and right can switch appearances with a change in registration of only one half of a halftone cell size.

Figure 6.38 shows experimental results for a typical xerographic printer using the three different configurations. In this lightness vs. chromaticity plot, the data measured for the rotated configuration are very close to the predicted model. The experimental results for the other cases show the same tendency as the model. The dot-on-dot gamut is smaller than the dot-off-dot

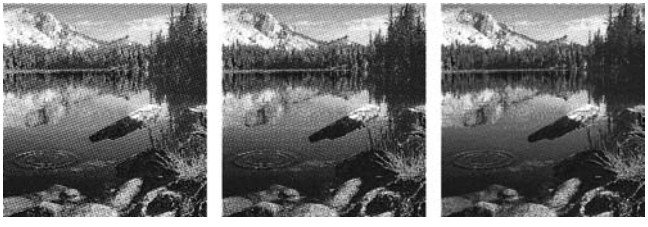


Figure 6A ***(See color insert)*** Natural color image in three orientations: dot-on-dot, rotated, and dot-off-dot. Examples show the change in color that can occur with a change in registration for different halftone orientations. The photographs on the left and right can switch appearances with change in registration.

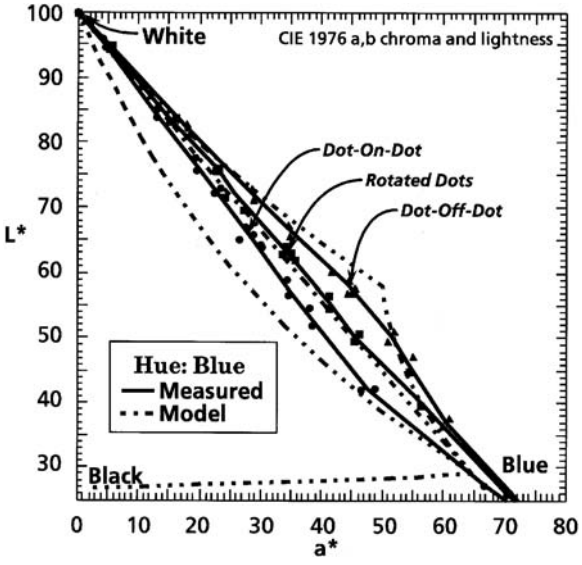


Figure 6.38 Experimental results for three halftone orientations.

results, with the rotated case lying between. That the two halftone configurations plot closer to the rotated dot results is likely due to the xerographic noise in the experimental prints. Photomicrographs of the prints show many isolated toner particles that appear random like the rotated dots.

The results of Rhodes and Hains have been confirmed by the theoretical and simulation studies of Gustavson.[27]

6.5.2.2 *Tone step uniformity*

One of the major effects of halftone orientation is the resultant distribution of tone levels. Figure 6.36 shows that the rotated dots have the most linear distribution of gray levels, while the dot-on-dot case has more levels bunched up in the highlights, and the dot-off-dot case has them bunched up in the high-chroma region. Nonuniformity in tone distribution can lead to tone-level contouring as well as lowering of the accuracy of the color correction operation.

6.5.2.3 *Sensitivity to moiré*

Of the three orientations, only rotated dots produce regular moiré by interaction between the screens. This moiré may be due to either angle separation and/or differences in screen frequency. In addition, a secondary moiré may result due to an interaction between a black 45° screen and the resultant beat between the ±15° magenta and cyan screens. Any or all of these effects can also interact with detail or scanning artifacts in an image. The other two cases can cause irregular low-frequency color moiré due to misregistration as the two cases reverse themselves.

Dot-on-dot can interact with image detail or scanning artifacts that come close to the single-screen frequency, but this interaction can be strong due to the high visual contrast of the darker dots. Dot-off-dot halftones have an extra chance for interaction due to the higher frequency that is produced when several colorants have similar values. The interaction may be weaker due to the lower visual contrast of the individual colorant screens.

6.5.3 *Recommendations*

If the colorants were "ideal," there would be no difference in gamut due to halftone orientation.[24,28] It is the unwanted density of the two colorants that causes the differences in color gamut with halftone orientation. The dot-off-dot configuration has a significantly larger gamut than the other two cases.

It is also apparent that the non-rotated cases are most sensitive to errors in registration. Also, equal halftone percentages do not produce equal steps in $L^*a^*b^*$. This means that digital halftone printers would require the ability to print a large number of gray levels to avoid tone contouring.

The discontinuity in the two-colorant dot-off-dot halftone at the 50% dot coverage area would be a problem in printers with a limited number of gray levels.

It is attractive to speculate on how one might take advantage of the dot-off-dot configuration's larger gamut. Figure 6.34 was made with only two colorants, where the 50% levels fit together tightly in a checkerboard pattern with no overlap and no white showing. For four-color printing, some compromise would be required. This method could avoid the rosettes of rotated dots as well as have the largest gamut. However, the change in color with change in registration could be a serious problem, particularly if the process were noise free. Therefore, the printer would have to be capable of precise registration. In the meantime, the use of rotated dots is the safest bet.

6.6 *Moiré*

In digital color printing, the color separations are halftoned and imaged independently. Because of this, there are many chances for very slight misregistration to occur from one separation to the next. Misregistration of halftone dots can produce unpredictable colors because of the overlap of colorants that have unwanted absorption in their spectra. Rotated halftone screens are normally used to gain insensitivity to misregistration.

The conventional strategy is to choose equally spaced angles for the more visible screens of black, cyan, and magenta and to insert the less visible yellow screen between two of the others, e.g., 45° black, 15° magenta, 75° cyan, and 0° yellow. This is the best strategy in most cases. In cases of severe two-color moiré, it may be desirable to use a nonclustered screen for the yellow.

6.6.1 Rosettes

Rotated screen printing produces a rosette pattern that, at best, is about twice as large as the pattern of an individual screen. The 30° angles are chosen to give the highest frequency rosette pattern between C, M, and K. The rosette itself is a moiré pattern, but one that forces all possible dot overlaps in the smallest area. Thus, the result is a stable average color that is less sensitive to registration. Accurate angles are required to avoid low-frequency beats that would display even larger patterns. Particular attention should be made in designing the screens so as to cancel out low-frequency beats.

The "strength" of the rosette structure also depends on the separation technique. The more black is used in highlights and midtones (when a high GCR level is used), the more pronounced the rosette structure will be. Conversely, the less cyan and magenta used in the shadows (when a high UCR level is used), the less visible the rosette structure will be.[29]

The rosettes are meant to be random:

> No matter how far you go from an exactly registered dot-centered rosette, you will theoretically never find another in exact register, because the tangents of the screen angles and the ratios of the distances between dots are irrational numbers. In practice, however, this does not apply because of accidental inaccuracies, and even in the theoretical pattern many of the rosettes are in register within the accuracy of visual examination.[23]

In digital halftoning, it is even more likely that the pattern repeats, as the screens are implemented with rational tangent angles — even with large supercells.

6.6.2 Dot center phase

The rosette structure can be designed with one of two aims, the *dot-centered* or the *hole-centered* strategy.

A dot-centered rosette pattern forms when the *dot* centers of the CMK screens are registered at the same point. The smallest black structure in the highlights has the size of $2P$.[30]

A clear-centered rosette pattern forms when the *hole* centers of the CMK screens are registered at the same point. The smallest black structure in the highlights has the size of $\sqrt{2} * P$.[30] Figure 6.39 is a blow-up showing the appearance of these two structures.

The two rosette structures have different characteristics. The dot-centered rosette has a bigger color gamut. This may be because there is less overlap of the cyan and magenta screens, so their unwanted absorption spectra have less of a chance to interact. It may also be because there is less white paper visible to desaturate the colors.

The clear-centered rosette is less visible in the highlights and midtones than the dot-centered rosette, due to the smaller structure.[29–31] The clear-

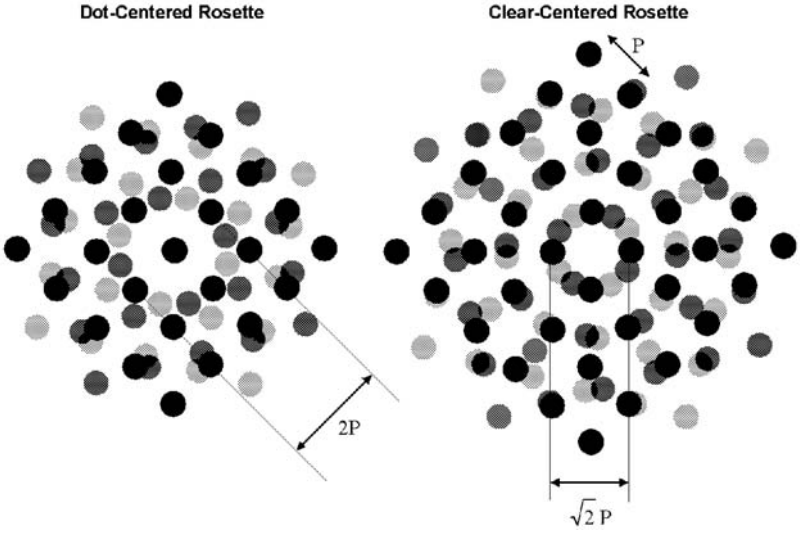


Figure 6.39 ***(See color insert)*** Rosette characteristics.

centered rosette also preserves the gradation better in the dark tones than the dot centered-rosette (results in a smoother TRC, perhaps less dot gain).[29,31]

Different rosette structures require different color calibrations. The clear-centered rosette tends to render neutral colors too green, and dot-centered rosette tends to too magenta. The color difference is the maximum between these two extremes among all rosette structures.[30]

A clear-centered rosette is converted into a dot-centered rosette, and vice versa, when the screens are digitally inverted. A clear-centered rosette can also change to a dot-centered rosette through many different screen phase changes, and vice versa. The phase changes may be due to alignment, registration, and moiré.

An example of a low-frequency moiré due to switching between clear-centered and dot-centered rosettes can be seen in Figure 6B (see color insert).†

6.6.3 *Two-color moiré*

Printing with digital halftone screens often produces moiré in color prints due to the approximate screen angles and frequencies that are used by the rational-tangent screen algorithm. The most common moiré is due to the use of the zero-degree screen for the yellow colorant. When used with the usual 15, 45, and 75° screens for cyan, black, and magenta, the yellow is only 15° away from the cyan and the magenta angles. The two-color moirés then

† Thanks to Dr. Robert P. Loce of the Xerox Corporation for his thoughts and his collection of moiré references, and to Shirley Cheng for the rosette figure.

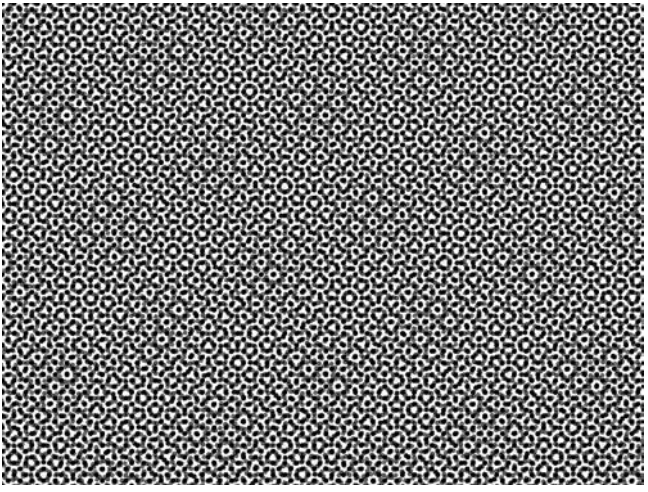


Figure 6B ***(See color insert)*** Four-color moiré. This figure illustrates the low-frequency moiré that results from the rosette switching between clear-centered and dot-centered phases.

appear most commonly in light greens, pinks, and tans. One method of avoiding this moiré is to change the frequency of the yellow screen.

> ...it appears that there is some justification for using a 133-line screen for yellow when the others are of 120-line screens.[32]

An excellent overview of the moiré phenomenon in color separation can be found in Amidror.[33]

6.6.4 *Three-color moiré*

Other than the two-color interactions and the drift between dot-centered and hole-centered rosettes, more complicated moiré patterns can develop due to the use of rotated screens.

Indeed, the prints made with classic non-digital rotated screens can also have a very low-frequency moiré. The following discussion refers to this more complex moiré. It is paraphrased from Tollenaar.[20]

> If we print the plates for cyan, black, and magenta with screen angles inaccurately set to separations of 15°, 45°, and 75° with respect to the yellow screen, we find a large checkered moiré, which is especially troublesome in the darker areas. At a first glance, this moiré is surprising, since the three mentioned screens vary in position every 30° and in that way should not be able to produce a visible moiré. However, let us attend only to the interference of the magenta and cyan printers, the screens of which are separated by 60°. The interaction of these two screens produces a structure upon the print that is invisible

to the naked eye but is present without any doubt. This structure is further oriented according to the bisector of the angle between the red and blue screens and, therefore, is parallel to the 45° line. Now, this structure interferes with the period of the black print, which has the same period and is printed at 45°. A small angular variation of one of the screens or plates causes, therefore, a moiré, which can be avoided only by locating very accurately the angles for the screens — at any rate, as long as one insists on the use of plates with uniform screens.

The pattern is large, because the angle between the black printer and the interfering moiré of the cyan and magenta printers is very small. In fact, it should be zero in an ideal print. Therefore, any small deviations of this angle or small dimensional changes in the paper during the printing procedure cause large changes in the appearance of the pattern.

In practice, a correctly made print does not show a regular pattern of this kind but, instead, very faint clouds of rosette-like groupings of screen dots, irregularly distributed all over the printed area. If the screen angles are not correctly set but deviate as little as one degree, the clouds arrange themselves into the regular pattern of very low frequency.

6.7 *Nonorthogonal halftone screens*

6.7.1 *Introduction*

Most halftone screens used in color reproduction are orthogonal screens or screens in rectangular shapes, more likely in squares. Nonorthogonal screens refer to screens in general parallelogram shapes. In 1970, Holladay developed an efficient way of encoding the halftone screens.[34] The method gives a unique halftone description with the advantages of simple implementation and a small memory requirement. The algorithm is based on geometry of general parallelograms; therefore, as explicitly pointed by Holladay, any nonorthogonal halftone screens can be produced and implemented by his method exactly as orthogonal screens. Obviously, orthogonal screens are only a small subset of the complete set of all nonorthogonal screens. However, using general nonorthogonal halftone screens had not shown any major advantage over orthogonal ones and, as a result, not much interest had been previously brought in the halftoning field until recently.[35,39] Wang, Fan, and Wen suggested using nonorthogonal screens for moiré-free color halftoning by searching integer equation solutions.[35]

In a typical screening process, a halftone screen is applied repeatedly in a way similar to a tiling process indicated by the gray lines in Figure 6.40. With any constant input, the halftone output by this halftone screen will be a two-dimensional spatial periodical function. From Fourier analysis, it is clear that the spectrum of the halftone output is composed of, and only of, the two fundamental frequencies and their higher-order harmonics from the

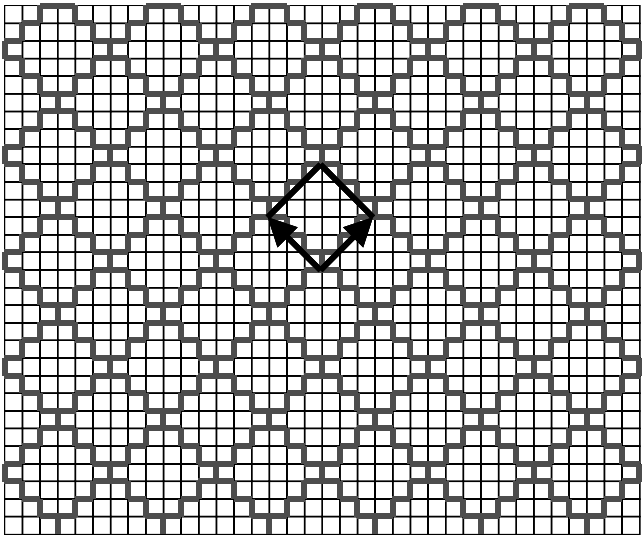


Figure 6.40 Digital grid with an orthogonal halftone screen outlined by red lines.

Fourier transform of the halftone screen. There is no component with a frequency lower than the two fundamentals. For most of moiré analysis in halftone screen design, we may concentrate on fundamental frequencies of halftone screens, because the dominating moiré is most likely due to the interaction between fundamental frequencies of individual halftone screens for different color channels. First, let us consider a typical rotated orthogonal screen, as shown in Figure 6.40. The geometry of this halftone screen can be specified by the dual representation of the Fourier transformation. In the spatial domain, the rectangular screen is specified by two orthogonal vectors, $\mathbf{v}_1$ and $\mathbf{v}_2$, shown in Figure 6.41; while in the Fourier transform domain, it is represented by two orthogonal frequency vectors, $\mathbf{V}_1$ and $\mathbf{V}_2$, shown in Figure 6.42. As properties of the Fourier transformation, the two frequency vectors, $\mathbf{V}_1$ and $\mathbf{V}_2$, are perpendicular to the two spatial vectors, $\mathbf{v}_1$ and $\mathbf{v}_2$,

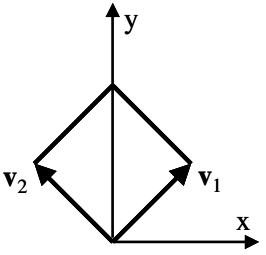


Figure 6.41 Spatial vector representation of an orthogonal halftone screen.

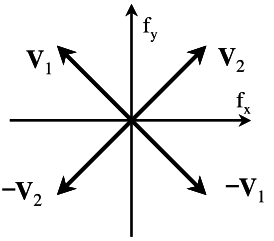


Figure 6.42 Frequency vector representation of an orthogonal halftone screen.

respectively, and the moduli, or the absolute values, $|\mathbf{V}_1|$ and $|\mathbf{V}_2|$, are equal to $1/|\mathbf{v}_2|$ and $1/|\mathbf{v}_1|$, respectively.

From discussions in Chapter 5 and elsewhere in this chapter, one may have learned that, to eliminate or reduce moirés caused by interaction between different color channels, it is often critical to have precise rotation angles of halftone screens. For example, to avoid three-color moirés, traditional analog halftoning uses an identical dot screen for cyan, magenta, and black channels with 15, 75 and 45° rotation, respectively. Unfortunately, in digital halftoning, the selection of possible rotations for halftone screens is greatly restricted by a digital grid, or raster, defined by the location of physical pixels. The tangent of the rotation angle, specified by the argument of the spatial vector $\mathbf{v}$, or the argument of the frequency vectors $\mathbf{V}$, has to be a rational number, because the two Cartesian-coordinate components of a spatial vector $\mathbf{v}$ have to be integers. For example, a 45° rotation of a halftone screen, as shown in Figure 6.40, may be achieved in digital halftoning, because $\tan(45°) = 1$. However, neither 15 nor 75° rotation of a halftone screen can be implemented digitally, because the tangents of these angles are irrational. Various digital halftoning methods have been proposed for precisely or approximately reaching certain desired rotation angles of halftone screens.[36–39] Perhaps the most popular approach is the supercell, which is a rational tangent screen composed of many smaller subcells that are not uniform in size and shape.[37] The drawback of supercells is that a large halftone screen contains low fundamental frequencies. If the fundamentals or their spatial harmonics fall in the range of the sensitivity function of the human visual system, additional effort has to be made during the screen design process to avoid possible low-frequency artifacts that might be shown in the halftone outputs. In general, designing a supercell is much more difficult than designing simple halftone screens with single or a few centers. On the other hand, the selection of small orthogonal screens with different rotation angles is quite limited, especially for devices with relatively low resolutions, which makes it almost impossible to find a moiré-free solution using rotated simple orthogonal halftone screens for color halftoning.

In the following sections, a general analysis on nonorthogonal screens will be discussed, which will lead to a method proposed by Wang et al.[35] for searching moiré-free solutions using simple nonorthogonal halftone screens.

6.7.2 *Dual representation of nonorthogonal screens*

A general nonorthogonal halftone screen is outlined by the red lines in Figure 6.43. The shape of this parallelogram screen can be specified by two vectors, $\mathbf{v}_1(x_1, y_1)$ and $\mathbf{v}_2(x_2, y_2)$, shown in Figure 6.44. The two fundamental frequencies of the Fourier transform of this screen can be represented by two frequency vectors, $\mathbf{V}_1(f_{x1}, f_{y1})$ and $\mathbf{V}_2(f_{x2}, f_{y2})$, shown in Figure 6.45. Similar to the orthogonal case, $\mathbf{V}_1$ and $\mathbf{V}_2$ are perpendicular to $\mathbf{v}_1$ and $\mathbf{v}_2$, respectively. However, the moduli of the frequency vectors $|\mathbf{V}_1|$ and $|\mathbf{V}_2|$ are not given by the reciprocals of $|\mathbf{v}_2|$ and $|\mathbf{v}_1|$ as for the orthogonal screens. Instead, $|\mathbf{V}_1|$ and $|\mathbf{V}_2|$ are equal to the reciprocals of h_1 and h_2, which are the heights, or the pitches, shown by the dot lines in Figure 6.44. Because the product $|\mathbf{v}_1| * h_1 = |\mathbf{v}_2| * h_2 = A$ is the area of the specified parallelogram, we may write the moduli of the frequency vectors $\mathbf{V}_1$ and $\mathbf{V}_2$ as the following equations:

$$|\mathbf{V}_1| = \frac{|\mathbf{v}_1|}{A} \tag{6.1a}$$

$$|\mathbf{V}_2| = \frac{|\mathbf{v}_2|}{A} \tag{6.1b}$$

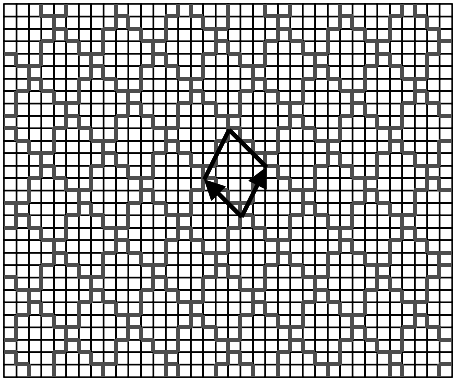


Figure 6.43 Digital grid with a nonorthogonal halftone screen outlined by red lines.

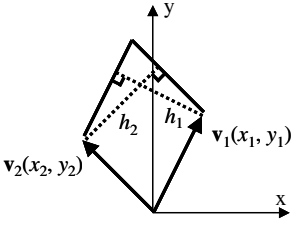


Figure 6.44 Spatial vector representation of a nonorthogonal halftone screen.

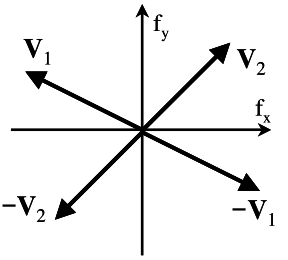


Figure 6.45 Frequency vector representation of a nonorthogonal halftone screen.

where A is given by the absolute value of the cross product of two vectors, $\mathbf{v}_1 \times \mathbf{v}_2$, i.e.,

$$A = |x_1 y_2 - x_2 y_1| \tag{6.2}$$

Because the spatial vector $\mathbf{v}_1(x_1, y_1)$ and the frequency vector $\mathbf{V}_1(f_{x1}, f_{y1})$ are perpendicular to each other, so are $\mathbf{v}_2(x_2, y_2)$ and $\mathbf{V}_2(f_{x2}, f_{y2})$, and from Equations 6.1a and 6.1b, it is not difficult to prove that

$$f_{x1} = \frac{-y_1}{A} \tag{6.3a}$$

$$f_{y1} = \frac{x_1}{A} \tag{6.3b}$$

$$f_{x2} = \frac{-y_2}{A} \tag{6.3c}$$

$$f_{y2} = \frac{x_2}{A} \tag{6.3d}$$

Under above-mentioned digital grid constraint (i.e., all spatial vectors are specified by integers in the Cartesian coordinate), the area A given by Equation 6.2 must be a nonnegative integer. Consequently, all frequency-vector components in the Cartesian coordinate are rational numbers. Imagine a parallelogram specified by the two frequency vectors $\mathbf{V}_1$ and $\mathbf{V}_2$; one may see that it would be a rotated and scaled version of the parallelogram halftone screen with 90° rotation and $1/A$ scaling. If the condition

$$\frac{x_1}{y_1} = \frac{-y_2}{x_2} \tag{6.4}$$

is satisfied, the general parallelogram becomes rectangular. Furthermore, if

$$x_1 = \pm y_2 \tag{6.5a}$$

and

$$y_1 = \mp x_2 \tag{6.5b}$$

the parallelogram becomes a square.

It is interesting to notice that changing one spatial vector, say $\mathbf{v}_1$, of an orthogonal screen will only affect the frequency $\mathbf{V}_1$, while changing $\mathbf{v}_1$ of a nonorthogonal screen will affect, in general, both frequencies, $\mathbf{V}_1$ and $\mathbf{V}_2$. Because the two spatial vectors can be specified independently for a parallelogram, the number of different frequencies by using nonorthogonal screens is approximately N^2, comparing with N by using orthogonal screens. In practice, it is possible to obtain many nonorthogonal screens that look just like orthogonal screens by choosing the two vectors with an angle between them close to 90°. This might be not difficult to implement, especially for devices with relatively high resolutions in either one or two dimensions. In addition, a diamond-shaped parallelogram can produce halftone outputs with a hexagon-like halftone texture, which might also be interesting.

6.7.3 *Moiré-free conditions*

One particular application of nonorthogonal halftone screens is to provide perfect solutions for moiré-free color halftoning. Moiré patterns may appear in the printed documents for several possible reasons. In color printing, the most unwanted moiré is due to the superposition of the halftone screens of the different process colors and due to "unwanted absorptions" in the colorants. If there are no unwanted absorptions, there is no moiré, as the colorants do not interact. Using Fourier analysis provided for halftone screens, we can express the result caused by superposition of two different colors as their frequency-vector difference, $\mathbf{V}_{cm} = \mathbf{V}_c \pm \mathbf{V}_m$, where $\mathbf{V}_c$ and $\mathbf{V}_m$ are two frequency components from two different colors, e.g., cyan and magenta, and $\mathbf{V}_{cm}$ is the difference vector. Because each Fourier component has a corresponding conjugate (i.e., there is always a frequency vector $-\mathbf{V}_c$ that represents the conjugate component of $\mathbf{V}_c$), the sign definition of frequency vectors is rather arbitrary. For each parallelogram screen, there are two fundamental frequency vectors; therefore, the color mixing of two screens for two different colors yields four difference vectors as illustrated by Figure 6.46. If any one of these difference vectors is much shorter than the cut-off frequency of the sensitivity function of human visual system and not very close to zero, there is a possibility that two-color moiré will appear on the halftone output at the frequency represented by the corresponding difference vector. The common strategy to avoid any two-color moiré is to

make sure that no two-color difference vector will be too small. The two-color moiré-free condition can be summarized by

$$|\mathbf{V}_c \pm \mathbf{V}_m| > V_{\min} \tag{6.6}$$

where $\mathbf{V}_c = \mathbf{V}_{c1}, \mathbf{V}_{c1}, -\mathbf{V}_{c2}, -\mathbf{V}_{c2}$; $\mathbf{V}_m = \mathbf{V}_{m1}, -\mathbf{V}_{m1}, \mathbf{V}_{m2}, -\mathbf{V}_{m2}$, and $V_{\min}$ is a frequency limit set at somewhere between 50 and 70 lpi for just-noticeable moirés.

It is well known that the most troublesome moiré is the three-color moiré, usually appearing as the cyan–magenta–black moiré in outputs by CMYK four-color printers. As an extension of the two-color case, the three-color moiré-free condition can be summarized by

$$|\mathbf{V}_c \pm \mathbf{V}_m \pm \mathbf{V}_k| > V_{\min} \tag{6.7}$$

where $\mathbf{V}_c = \mathbf{V}_{c1}, -\mathbf{V}_{c1}, \mathbf{V}_{c2}, -\mathbf{V}_{c2}$; $\mathbf{V}_{\mathrm{m}} = \mathbf{V}_{m1}, -\mathbf{V}_{m1}, \mathbf{V}_{m2}, -\mathbf{V}_{m2}$; $\mathbf{V}_k = \mathbf{V}_{k1}, -\mathbf{V}_{k1}, \mathbf{V}_{k2}, -\mathbf{V}_{k2}$, and $V_{\min}$ is set similarly to the two-color case. Because there are, all together, 16 different combinations of different color components, in practical terms, it is difficult to make all three-color difference vectors, as well as all two-color difference vectors, large enough to avoid any color moiré unless the halftone screens have very high-frequency fundamentals — say, higher than 200 lpi. An common alternative approach is to make two of the three-color difference vectors null while keeping rest large. Given that both the signs and the indices of frequency vectors are defined somewhat arbitrarily, without losing the generality, the three-color moiré-free condition can be specified by the following two vector equations:

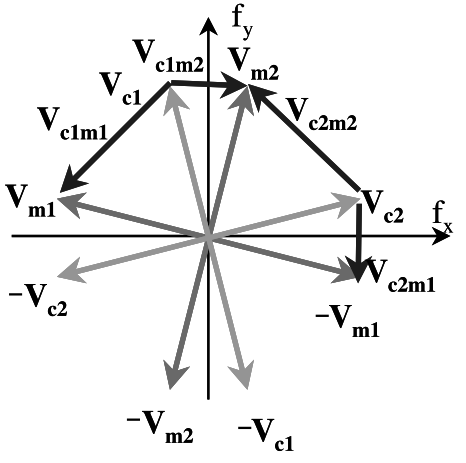


Figure 6.46 Difference vectors by interaction of two colors with different halftone screens.

$$\mathbf{V}_{c1} + \mathbf{V}_{m1} + \mathbf{V}_{k1} = 0 \tag{6.8a}$$

and

$$\mathbf{V}_{c2} + \mathbf{V}_{m2} + \mathbf{V}_{k2} = 0 \tag{6.8b}$$

It is not difficult to prove that, once Equations 6.8a and 6.8b are satisfied, the rest of the combinations of three-color components are equal to linear combination of higher-order harmonics from two colors. In most practical applications, they will satisfy the inequality of Equation 6.7. An example is illustrated by Figure 6.47.

Using scalar representation of Equations 6.3a through 6.3d, one can easily rewrite the moiré-free condition as follows:

$$\frac{x_{c1}}{A_c} + \frac{x_{m1}}{A_m} + \frac{x_{k1}}{A_k} = 0 \tag{6.9a}$$

$$\frac{y_{c1}}{A_c} + \frac{y_{m1}}{A_m} + \frac{y_{k1}}{A_k} = 0 \tag{6.9b}$$

$$\frac{x_{c2}}{A_c} + \frac{x_{m2}}{A_m} + \frac{x_{k2}}{A_k} = 0 \tag{6.9c}$$

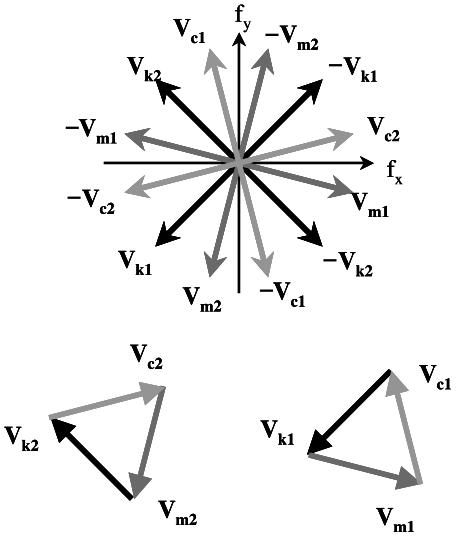


Figure 6.47 Three-color moiré-free conditions illustrated by two triangles formed by vector summations.

and

$$\frac{y_{c2}}{A_c} + \frac{y_{m2}}{A_m} + \frac{y_{k2}}{A_k} = 0 \tag{6.9d}$$

where the values of parallelogram area are given by Equation 6.2 or rewritten as

$$A_c = |x_{c1}y_{c2} - x_{c2}y_{c1}| \tag{6.10a}$$

$$A_m = |x_{m1}y_{m2} - x_{m2}y_{m1}| \tag{6.10b}$$

$$A_k = |x_{k1}y_{k2} - x_{k2}y_{k1}| \tag{6.10c}$$

Alternatively, two vector equations in spatial vectors can be derived from Equations 6.9a through 6.9d, i.e.,

$$\frac{\mathbf{v}_{c1}}{A_c} + \frac{\mathbf{v}_{m1}}{A_m} + \frac{\mathbf{v}_{k1}}{A_k} = 0 \tag{6.11a}$$

and

$$\frac{\mathbf{v}_{c2}}{A_c} + \frac{\mathbf{v}_{m2}}{A_m} + \frac{\mathbf{v}_{k2}}{A_k} = 0 \tag{6.11b}$$

If all Cartesian components of spatial vectors are integer specified, areas A_c, A_m, and A_k are also integers. So, by rearranging Equations 6.9a through 6.9d, the three-color moiré-free condition can be fully specified by the following four integer equations:

$$x_{c1}A_mA_k + x_{m1}A_kA_c + x_{k1}A_cA_m = 0 \tag{6.12a}$$

$$y_{c1}A_mA_k + y_{m1}A_kA_c + y_{k1}A_cA_m = 0 \tag{6.12b}$$

$$x_{c2}A_mA_k + x_{m2}A_kA_c + x_{k2}A_cA_m = 0 \tag{6.12c}$$

and

$$y_{c2}A_mA_k + y_{m2}A_kA_c + y_{k2}A_cA_m = 0 \tag{6.12d}$$

Obviously, other moiré-free conditions by the inequalities of Equations 6.6 and 6.7 can be also converted to integer inequalities in a similar manner.

6.7.4 Searching for moiré-free solutions

From the previous section, it is apparent that, although the original moiré analysis is mostly based on frequency calculation, all moiré-free conditions for nonorthogonal screen halftoning can be completely specified by spatial vectors, which define the shape and size of parallelograms. Furthermore, for digital halftoning, all these moiré-free conditions can be stated by either integer equations or integer inequalities. Therefore, the number of moiré-free solutions is finite, and all solutions can be searched by a computer, even though there are not enough equations and/or inequalities for analytic solutions. We have also learned that, due to the digital grid constraint, the choices of different parallelograms are quite limited, which, on the other hand, allows us to do a quick search for possible combinations for moiré-free color halftoning. For example, if we restrict the halftone screens to a frequency range above 120 lpi for a 1200×1200 dpi printer, the search is limited to all integers less than 10. Briefly, a possible search routine might be described by the following steps:

1. Search all possible parallelograms that meet the required screen sizes and shapes, and store them as two integer-specified spatial vectors, ($\mathbf{x}_1$, $\mathbf{y}_1$; $\mathbf{x}_2$, $\mathbf{y}_2$) into a screen list.
2. For each set of three parallelograms, say ($\mathbf{x}_{c1}$, $\mathbf{y}_{c1}$; $\mathbf{x}_{c2}$, $\mathbf{y}_{c2}$), ($\mathbf{x}_{m1}$, $\mathbf{y}_{m1}$; $\mathbf{x}_{m2}$, $\mathbf{y}_{m2}$), and ($\mathbf{x}_{k1}$, $\mathbf{y}_{k1}$; $\mathbf{x}_{k2}$, $\mathbf{y}_{k2}$), from the stored screen list, check
   a. Whether three-color moiré-free conditions, integer Equations 6.12a through 6.12d, are satisfied
   b. Whether all other two- or three-color moiré-free conditions, integer inequalities derived from Equations 6.6 and 6.7, are satisfied
   c. Whether other possible additional constraints, e.g., symmetry appearance, are satisfied
3. Save the result if all constraints are satisfied, otherwise, stop checking and continue to the next combination.
4. Evaluate the result. If necessary, change the requirements for screen size, shape, moiré frequency limit, or other factors, and redo the search.
5. Conduct a final evaluation by designing halftone screens based on selected geometries and by generating and printing testing patterns.

Of course, the above description provides only a skeleton of a possible search routine. Many possible variations might be implemented to speed up the search and/or meet additional requirements.

6.7.5 An example of moiré-free nonorthogonal screens

If we assume that all three nonorthogonal screens have the same area, i.e., $A_c = A_m = A_k = A$, integer Equations 6.12a through 6.12d can be significantly simplified, and the search for moiré-free solutions could be done even with-

out a computer. As an example, by selecting $A = 60$, a possible moiré-free solution is given by

$$\mathbf{v}_{c1} = (8, 2), \mathbf{v}_{c2} = (-2, 7)$$

$$\mathbf{v}_{m1} = (2, 7), \mathbf{v}_{m2} = (-8, 2), \text{ and}$$

$$\mathbf{v}_{k1} = (6, 5), \mathbf{v}_{k2} = (-6, 5)$$

The three parallelograms and their spectral representations are shown in Figure 6.48. When this set halftone screens is applied to a color printer with 1200 × 1200 dpi resolution, the halftone output will have the following frequency properties:

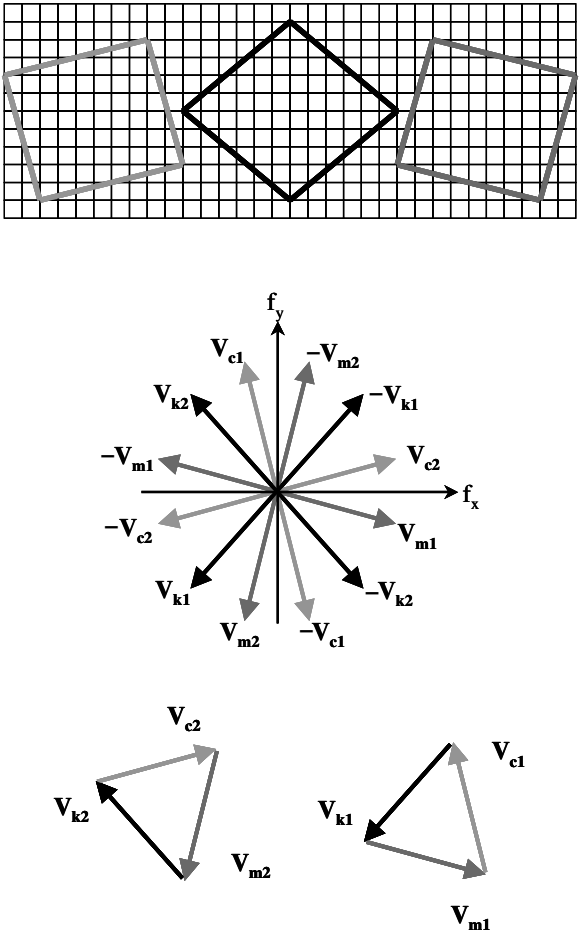


Figure 6.48 Three nonorthogonal halftone screens satisfying moiré-free conditions.

Cyan	$\tan^{-1}(8/2) \approx 75.96°$, 164.9 lpi and $\tan^{-1}(-2/7) \approx 15.95°$, 145.6 lpi
Magenta	$\tan^{-1}(2/7) \approx 15.95°$, 145.6 lpi and $\tan^{-1}(-8/2) \approx 75.96°$, 164.9 lpi
Black	$\tan^{-1}(6/5) \approx 50.2°$, 156.2 lpi and $\tan^{-1}(-6/5) \approx 50.2°$, 156.2 lpi

where the angle is defined as $\tan^{-1}(x/y)$, and x and y are the two integer components of the spatial vector $\mathbf{v}(x, y)$.

Figure 6.49 shows the output by this set of nonorthogonal halftone screens with certain constant input for cyan, magenta, and black channels. There are two scaled-up versions of the output with different scale factors in Figure 6.49. It is interesting to notice that the rosette pattern generated by this set of nonorthogonal screens has clear repeated structure, which is quite different from rosette patterns generated by classical rotated orthogonal screens.

6.8 FM halftoning methods

6.8.1 Introduction

There are several FM halftoning algorithms (see Section 6.2). Of these, error diffusion is the easiest FM halftoning method to implement. It also produces very high image quality, due to its low noise and inherent edge enhancement. This section will describe the error diffusion algorithm in detail, along with a threshold array emulation of error diffusion, stochastic screens.

Error diffusion was invented by Floyd and Steinberg in 1975. It was a simple algorithm created to improve the display of their images.[40] They did not understand why, but the algorithm produced very high-quality images, especially as compared with other algorithms available at the time.

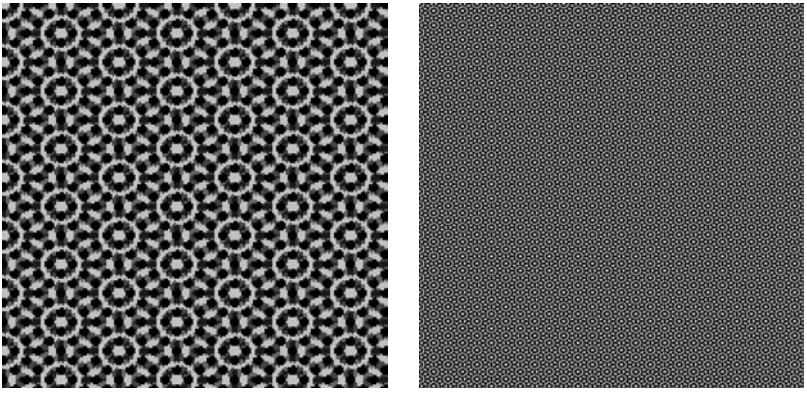


Figure 6.49 ***(See color insert)*** Sample halftone outputs by a set of moiré-free nonorthogonal halftone screens displayed in different scales.

Much work has been done to analyze and understand the workings of error diffusion[41] over the 25+ years since its invention. We now understand that much of its detail rendition comes from an inherent edge enhancement[42] built into the algorithm. An understanding of the errors generated[43] and what limits to place on them has developed. New filters[44] have been designed to limit texture artifacts. Extensions to color and vector error diffusion have deepened our understanding of the nature of the algorithm. Error diffusion continues to be one of the best digital halftoning algorithms available for image quality, detail rendition, and pleasing textures.

6.8.2 Error diffusion algorithm

Error diffusion is an adaptive thresholding algorithm that converts each pixel in the image to one of two levels, either black or white. Because the original pixel was not binary, an error was made in this process. This error is diffused to the neighboring pixels that are still to be processed, hence the name of the algorithm.

Figure 6.50 illustrates the algorithm part of the way through an image. The algorithm processes the pixels in an image sequentially, typically addressing the pixels from left to right and top to bottom. The pixels in the top half of the image have already been converted to black or white, and the pixel in the middle has just been converted to white. Because this pixel was originally gray, an error was made by setting it to white. The total picture is now too bright, by the difference of the input and output values of this pixel.

To correct for making the picture too bright, the error is divided into a small number of parts, which are then subtracted from the neighboring pixels that have not yet been processed. The original distribution of four parts used by Floyd and Steinberg is shown in Figure 6.51. Floyd and Steinberg chose this distribution because it gave them the most even distribution of textures in the output image with the fewest artifacts. Larger

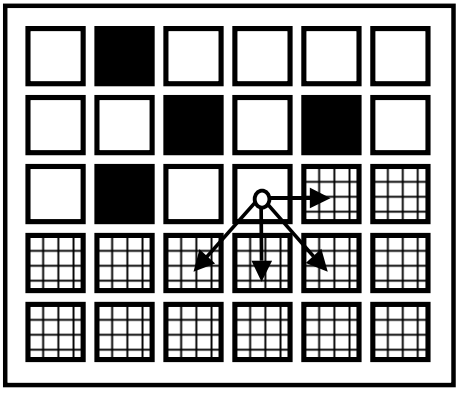


Figure 6.50 Error diffusion algorithm.

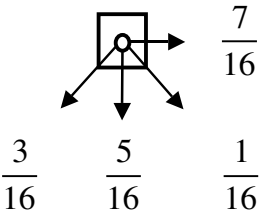


Figure 6.51 Floyd and Steinberg error distribution weights.

error arrays[45] and longer error arrays[44] also have been tried, yielding slightly better results.

Having modified the continuous-tone values of the neighboring pixels, the average brightness of the total image is now returned to its original value. In the process, however, the values of the remaining continuous-tone pixels have been changed. As the algorithm moves to threshold the next pixel in the middle scan line, that pixel's value will be different from its original value and may be closer to either black or white. That change will make this pixel more or less likely to be converted to either black or white.

When the algorithm has finished, all of the pixels in the output image will be either black or white. The only thing that the algorithm has guaranteed is that the total sum of all of the binary pixels will be very close to the total sum of the pixels in the original image. It turns out that the algorithm also guarantees that local averages of the binary image also are very close to the local averages of the corresponding regions of the original grayscale image. To understand how this occurs and why error diffusion has such good tone reproduction requires an analysis of the equations governing its behavior.

The diagram outlining this algorithm is shown in Figure 6.52. The input image is $i(x, y)$ and the output binary image is $b(x, y)$. The thresholding step is represented by the nonlinear function, $q[]$, and the filter, a_{jk}, represents the error distribution weights, such as those shown in Figure 6.51.

The output image does not have to be binary. If the nonlinear function, $q[]$, is a quantization rather than a thresholding step, then a multilevel output

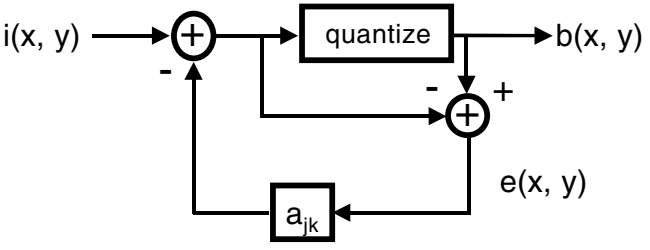


Figure 6.52 Schematic diagram of the error diffusion algorithm.

image will result. In general, the output can be any number of levels, but the basic assumption of the halftoning process is that the number of output levels is smaller than the number of input levels.

6.8.3 Error diffusion equations

The error diffusion algorithm can be described by two coupled, nonlinear equations (see Equations 6.13 and 6.14. Because of the nonlinearity, a closed-form solution of the output binary image, as a function of the input image, is not possible. The algorithm described in the previous section produces an empirical solution to these equations.

To simplify the discussion, white will be assumed to have a value of 1, and black is 0. In this case, both the input and output images can vary between 0 and 1 but cannot go outside those bounds. The errors are bound between $\pm 1/2$.

With these conditions, the error diffusion algorithm can be described by the following equations:

$$b(x, y) = q[i(x, y) - \Sigma(a_{jk}\, e(x - j, y - k))] \tag{6.13}$$

$$e(x, y) = b(x, y) - i(x, y) + \Sigma\, (a_{jk}\, e(x - j\Delta x, y - k\Delta y)) \tag{6.14}$$

where the summations are carried out over indices j and k, covering a small neighborhood of the current pixel. In the first equation, the value of the output image at every pixel, $b(x, y)$, is the thresholded value of the modified image pixel value at that point. The threshold function, $q[]$, compares its argument to $1/2$ and outputs white if the argument is greater than $1/2$, and outputs black otherwise. The modified input value is the difference between the original value of the pixel, $i(x, y)$, and the weighted errors from the neighboring pixels processed earlier. The weights are assumed to sum to unity.

The second equation defines the error at each pixel location, $e(x, y)$. This error is simply the difference between the output pixel value and the modified input pixel value. It is this error that is weighted and distributed to the neighboring pixels in the first equation. The error, $e(x, y)$, is a two-dimensional function and can be viewed as an image.[46]

If the nonlinear thresholding step is generalized to a quantization function, then multilevel halftoning is enabled. The more general quantization function, $q[]$, compares its argument to a series of levels and outputs the value of the level that is closest to the argument. All other parts of the two equations remain the same.

6.8.4 Spectral analysis of error diffusion

Although the two equations cannot be solved together, the error equation, which is a linear equation, can be further analyzed by using Fourier trans-

forms. After Fourier transforming Equation 6.14 and collecting all of the error terms together, it becomes

$$B(u, v) = I(u, v) + F(u, v)\ E(u, v) \tag{6.15}$$

where the filter function, $F(u, v)$ is given by

$$F(u, v) = 1 - \Sigma\ (a_{jk} \exp(-i(j\Delta x + k\Delta y))) \tag{6.16}$$

In these equations, capital letters are used to represent the Fourier transforms of the corresponding image functions.

Although Equation 6.15 still contains two unknown functions, $B(u, v)$ and $E(u, v)$, it brings to light an amazing relationship between them. Equation 6.15 states that the spectrum of the output image, $B(u, v)$, is identically equal to the input image spectrum, $I(u, v)$, plus a filtered version of the error spectrum, $E(u, v)$. Because the image spectrum is the desired output spectrum, the only source of artifacts or halftone textures in the output image must come from the error spectrum.

The error spectrum is still an unknown but, independent of its nature, an important conclusion from Equation 6.15 is that the contribution of the error spectrum to the output image has to pass through the filter function, $F(u, v)$. That means that any texture artifacts in the output image also must pass through this filter function. It is important, therefore, to understand the nature of this filter.

From Equation 6.16, it can be seen that the filter function, $F(u, v)$, depends on the error distribution weights, a_{jk}. In Figure 6.53, the magnitude of the filter function, $F(u, v)$, is shown for the Floyd and Steinberg weights. In this figure, the zero frequency point is in the center, and the corners are located at $\pm 1/2$ the scan frequency, or the Nyquist frequency. Larger filter values are shown as brighter values in the image.

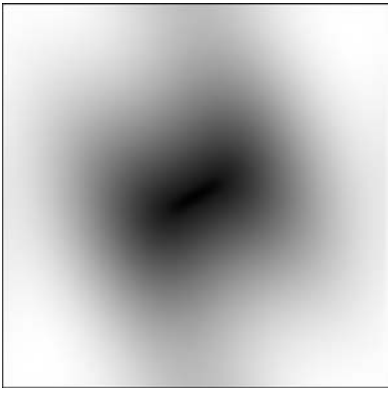


Figure 6.53 Error diffusion filter for the standard error weights.

There are two clear observations to be made from the image in Figure 6.53. The first is that the filter function, $F(u, v)$, is a highpass filter. It blocks DC values and passes the high spatial frequencies. The second observation is that the filter is asymmetric. It suppresses the low spatial frequencies in one direction much more strongly than it does in the perpendicular direction.

These two observations lead to two very important conclusions about error diffusion. The first conclusion is that the halftone textures in the error diffusion image contain no very low spatial frequencies. The textures must come from the error spectrum, and the error spectrum must pass through a high-pass filter. Therefore, the output halftone image is made of only high-spatial-frequency halftone textures. This is the origin of the well known blue noise characteristics of error diffusion.[41]

The second conclusion is that these textures are asymmetric. The filter function suppresses texture frequencies oriented roughly at –55°. This enhances texture frequencies perpendicular to that direction or at roughly 35°. This is the origin of the "worms" in the highlights and shadows.

In the highlights, the gray level is very close to white. As an FM halftoning algorithm, a highlight level is represented as a few widely spaced black pixels in a white field. This creates a lower-spatial-frequency halftone pattern, which is passed through the asymmetric part of the filter function. The result is a pattern that is closely spaced in one direction and widely spaced in the perpendicular direction, i.e., lines of black pixels, or "worms." The same effect happens in the shadows, where a few white pixels on a black background form directional lines. The many attempts to fix the problem of the "worms" by altering the way the error are distributed are described elsewhere.[47]

6.8.5 *Error image and edge enhancement*

Another distinct advantage of error diffusion over other halftoning algorithms is that it inherently builds edge enhancement into the output halftoned image. Error diffusion images look sharper than images halftoned with other techniques, because they have been edge enhanced.

The edge enhancement arises from the influence of the error image, $e(x, y)$. Although the error image cannot be predicted from the input image and the error weights, the error image can be calculated empirically by modifying the error diffusion algorithm to save the errors as the algorithm progresses.

An example of an error diffusion image and its error image are shown in Figure 6.54. It is clear from this figure that the error image looks quite a bit like the input image. A correlation measurement of the two images suggests that there is a linear component of about 1/2 of the continuous-tone input image in the error image, for the standard Floyd and Steinberg weights. There are different amounts for other error distributions.[46]

The linear component of the input image in the error image can be expressed as

Figure 6.54 Error diffusion image and its corresponding error image.

$$E(u, v) \sim 1/2\ I(u, v) + R(u, v) \tag{6.17}$$

where $R(u, v)$ is the spectrum of the remainder of the error image after the linear component is subtracted. To the eye,[46] this remainder error image appears to be uncorrelated to the original image. If Equation 6.17 is substituted into Equation 6.15, the output spectrum becomes

$$B(u, v) = I(u, v)\ (1 + 1/2\ F(u, v)) + F(u, v)\ R(u, v) \tag{6.18}$$

From this equation, it can be seen that the output spectrum is related to an input spectrum that has been enhanced by the addition of a high-pass term that boosts the high spatial frequencies. This is the origin of the inherent edge enhancement in error diffusion.

6.8.6 *Color error diffusion*

In the analysis presented so far, only monochrome images have been considered, and it does not directly apply to color input images. The introduction of color into error diffusion changes the nature of the algorithm along with the visual characteristics of the output image. A different error diffusion algorithm is required to halftone a color input image. At the time of this writing, research on color error diffusion is ongoing, and it is clear that new and improved algorithms will be continue to be developed in the future.

The simplest color error diffusion algorithm, and the first one developed, is color scalar error diffusion. Although each pixel in a color image is actually

a three-element color vector, a scalar method can be applied to each color separation independently. The scalar method used is just the monochrome error diffusion method described earlier. For display applications, the pixel values remain in RGB color space but, for printing applications, the pixel values are first converted to CMY or CMYK.

Error diffusion is known as a high-spatial-frequency halftone method that preserves detail and reduces halftone texture. This is still true of each of the three-color separations of the output scalar color error diffusion image. This is no longer true, however, of the combined output color image. When the three separations are processed independently, the output pixels at any given location are uncorrelated with the pixels at the same location in the other separations. As a result, color errors and low-spatial frequencies are introduced as the three separations are combined together in the output image.

The interaction of three uncorrelated error diffusion patterns can be seen in the monochromatic combination of three light gray error diffusion patterns in Figure 6.55. The three patterns, shown on the right in the figure, are made from gray patches at three slightly different levels. The three error diffusion patterns are all high-spatial-frequency patterns that have little low-spatial-frequency content.

The image on the far left of the figure is the union of the three patterns; i.e., at each pixel, if one of the three patterns is black, then this pixel is turned black. This union is a simulation of the effect of combining the three patterns in color and is only a rough measure of the luminance of the resultant image. From the left image, it can be seen that a strong low-spatial-frequency noise component is introduced into the texture pattern. In a color image, this pattern is less pronounced but is still objectionable.

The noise pattern is introduced because, when the three patterns are combined together, the uncorrelated arrangements of pixels sometimes overlap and sometimes lie adjacent to each other. The individual pixels from the three separations are different colors, i.e., cyan, magenta, and yellow. Because inks have unwanted absorptions,[48] different combinations of the same colored pixels produce different colors. This introduces low-spatial-frequency color errors in addition to the luminance errors already mentioned. These errors in color reproduction cannot be removed by calibration, and they make scalar color error diffusion unacceptable for color printing. There are

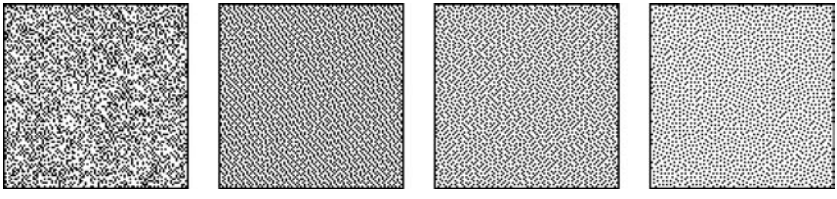


Figure 6.55 Combination of three uncorrelated error diffusion patterns.

no unwanted absorptions on color displays, but the low-spatial-frequency luminance patterns are still present.

6.8.7 *Vector error diffusion*

Vector error diffusion was developed to extend error diffusion to color displays with a limited palette. Scalar error diffusion gave very poor quality on such displays, while vector error diffusion greatly expanded the color resolution of the displayed image. Later, vector error diffusion was also applied to printing to solve the problem of the errors in color reproduction that are introduced by scalar error diffusion. Although vector error diffusion solved both the problem of palette imaging and random color errors, it introduced its own problems related to runaway errors. Despite its problems, in general, vector error diffusion produces a higher quality image than does scalar error diffusion.

The vector error diffusion algorithm is a very simple vector extension of standard error diffusion. The first step in that extension is to choose a set of primary colors. These primaries can be anywhere in the allowable color space. In scalar error diffusion, the primaries correspond to the maximum and minimum allowable values; for example, black and white in the monochrome case.

The basic extension of vector error diffusion is to generalize the quantization function, $q[]$, in Equation 6.1 to a vector choice. In the monochrome case, the quantization function is either a threshold, for black and white, or a quantization to a smaller number of levels for the multilevel case. In either case, one matches the input level to a set of allowable output levels and chooses the closest output level. In vector error diffusion, the quantization function measures the vector distance from the input color to all of the allowable primaries and chooses the primary that has the shortest distance.

$$q[i(x,y)] = \arg\min \|i((x,y) - p_k)\| \tag{6.19}$$

The $\|\cdot\|$ symbol represents the Euclidean distance measure of the difference vector argument, and the "arg min" function chooses the value of p_k that minimizes the distance measure. The choice of the color space in which to measure the error distance has a great influence on the performance of the algorithm. A visually uniform color space, such as $L^*a^*b^*$, is typically used to make the error measurements related to visual perception.

In the case where the errors are measured in the same color space as the output values, and the primaries are the eight corners of the color cube, the vector calculation produces the identical result as the scalar calculation. Vector error diffusion produces a different result when nonlinearity is introduced by the choice of the color space or of unevenly spaced primaries.

Vector error diffusion was first introduced to deal with color monitors.[49] Early displays, and even some contemporary ones, operate with a reduced number of colors, e.g., 256. These graphics boards have an 8-bit lookup table

that provides a palette of 256 colors, which can be arbitrarily chosen out of the full range of 24-bit colors. Color displays are a special case in which multiple levels can be shown and for which RGB is an appropriate color space to use. Even so, the limit of 256 colors is a severe restriction that greatly hampers the quality of the output image when produced by scalar error diffusion.

Although it may seem that 256 colors is a very small number of colors, it is much larger than the 8 primary colors that the binary scalar error diffusion would use. This enables the vector error diffusion algorithm to be multilevel, thereby reducing the visibility of the texture patterns that are produced by a proper choice of additional primary colors. The $L^*a^*b^*$ color space was shown[50] to be a suitable choice for choosing visually uniformly spaced colors, after an adjustment that more heavily weights the luminance. Methods have also been studied[51] to create universal palettes that can be used on more than one image simultaneously.

More recently, groups have begun to study how to adapt vector error diffusion to printed images. The print medium is significantly different from displays because of the nonlinearity introduced by unwanted absorptions of the inks. The techniques to address color calibration issues are covered elsewhere in this book, in the discussions on colorimetric reproduction.

The goal of applying vector error diffusion to printed images is to address the problem of errors in color reproduction caused by the interactions of uncorrelated error diffusion patterns in scalar error diffusion. By calculating the color errors directly from the vector differences, the color differences in the output image can be minimized. In vector error diffusion, the texture patterns in the CMYK separations are well correlated, so low-spatial-frequency texture patterns are minimized, producing less visual texture noise to the eye.

The use of $L^*a^*b^*$ color space to measure the color differences can help choose less-visible alternative patterns that scalar error diffusion cannot choose. For example, it has been pointed out[52] that a gray level can be represented either by an alternating pattern of black and white pixels or by a higher spatial frequency pattern of cyan, magenta, yellow, and white pixels. The latter combination of colored pixels has a lower luminance contrast and would be more likely chosen by the color difference quantization function in Equation 6.19. The tendency to choose lower contrast and higher spatial frequencies means that vector error diffusion produces much higher quality images than does scalar error diffusion.

This higher quality does not come without problems, however. Two problems arise as a consequence of measuring errors in $L^*a^*b^*$ color space while printing the output colors in a different color space, such as CMY. To achieve the proper combination of colored pixels that reproduce the input color, the feedback algorithm in error diffusion causes the production of very large errors. Because the errors are large, it can take many pixels to reach the equilibrium position. Also, at input color transitions, the accumulated error distorts the new color causing a colored smear over the transition

boundary. These two problems[53] are called *slow response* and *color smear.* Methods to limit the errors within the algorithm have been developed[53,54] to solve these problems.

6.8.8 Semi-vector error diffusion

A new algorithm, called semi-vector error diffusion, was recently proposed[55] that combines the best parts of both scalar and vector error diffusion while avoiding the major problems. Semi-vector error diffusion is designed for printing applications, and it is intended for CMY images. Any color correction is done before error diffusion is performed. The output images have very good image quality and are close to those achieved with vector error diffusion.

The algorithm is simple. It starts by applying standard scalar error diffusion to the yellow component. It turns out that yellow has the least brightness change and also the least unwanted absorptions. For all practical purposes, the yellow component is visually independent of the cyan and magenta components. If this is not the case, the yellow component can be processed with the other two components in a slightly more complicated sum and difference algorithm.[56]

The cyan and magenta separations are now quantized together using the following rules, as shown in Figure 6.56. First, because CMY are darkness-increasing components, a value of 1 means full ink, and a value of 0 means white, or no ink. If the sum of cyan and magenta is greater than 1.5, then set the output color to blue. If the sum is less than 0.5, then set the output color to white. Otherwise, set the output color to cyan or magenta, depending on which is larger. The cyan and magenta errors are calculated separately and diffused within their independent separations.

The semi-vector algorithm is a clear mixture of the scalar and vector algorithms. The error handling is done independently, as in the scalar method, but the quantization is calculated with the cyan and magenta separations simultaneously, as in the vector algorithm.

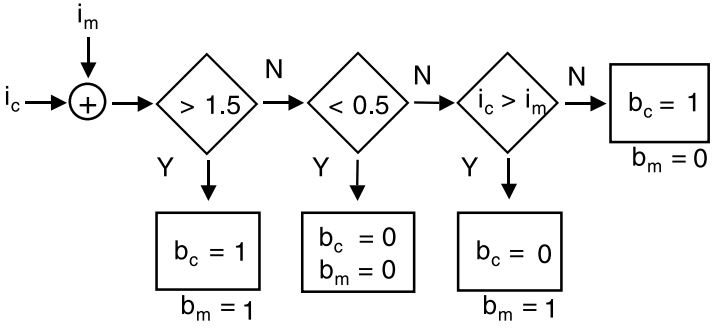


Figure 6.56 Schematic diagram of semi-vector quantization.

Because the errors are calculated in the scalar manner, the errors are very stable, and they do not exhibit runaway modes as the errors do in vector error diffusion. As a result, semi-vector does not exhibit the problems of slow response or color smear.

Because the cyan and magenta separations are calculated together, the error diffusion patterns in the two separations are well correlated, and the combination contains high-spatial-frequency textures. Semi-vector error diffusion, therefore, avoids the random color errors and low-spatial-frequency patterns that scalar error diffusion exhibits.

Finally, semi-vector error diffusion is a simple algorithm that avoids the complexity and computation of vector error diffusion while providing a high level of image quality with high spatial frequencies and few artifacts.

6.8.9 Stochastic screens

In general, FM halftoning methods, such as error diffusion, pulse density modulation, etc., are spatially adaptive algorithms. These methods adjust the output pixel value based not only on the immediate input pixel value but also based on neighboring input and output values. It is because of the adaptive nature of the FM halftoning algorithms that they can achieve the high-spatial frequencies and the high image quality that they do.

A stochastic screen is a standard threshold array halftoning screen that is designed to produce an output halftoned image that looks like an error-diffusion image. The application of the screen is not done in a spatially adaptive method but rather as a point process that does not depend on the neighboring values of the input image. The texture patterns in an image halftoned with a stochastic screen are made up of pixels that are randomly dispersed in the image in a high-spatial-frequency manner. The spatial frequencies of the screen have been shaped so that the output textures are similar to those produced by error diffusion.

The trick in a stochastic screen is in the design of the screen. This design step requires a spatially adaptive algorithm that iterates many times to search for optimal solutions. Constraining the threshold array itself to high spatial frequencies[57] is not sufficient. The output halftone patterns at many levels need to be constrained to contain high spatial frequencies.

The blue noise mask[58] is a stochastic screen that is constructed by constraining the output halftone patterns. The pattern for a given level is designed by an iterative process that starts with a random seed pattern. This pattern is blurred with a filter whose width is narrowest at mid-gray. The rms error between the pattern and the seed is calculated. The errors are sorted by value and a set of M pixels with the largest opposite errors are swapped. A new rms error is calculated, and the M swaps are accepted if the rms error decreases. An example of an initial seed pattern, the errors generated by blurring, and the final pattern produced after many iterations are shown in Figure 6.57.

After the pattern for a given level is determined, the seed pattern for the next level is generated by adding M random nonoverlapping, black pixels to it, where M is the number needed to achieve the next average gray level. The process is repeated at this level to generate a new pattern. The threshold array is generated by filling each pixel location with the gray level value at which the pixel turns from black to white.

Other methods of generating stochastic screens have been developed using visual metrics[59] and genetic optimization.[60] All of these have the property that individual gray levels are first optimized and then combined to create the threshold array.

There is one method for generating a stochastic screen that works directly with the threshold array values.[61] This method starts by generating a random array of threshold values. A cost function is computed that penalizes pairs of thresholds that will generate pixels of the same color that are close together. This cost function is computed over all gray levels and combined into one result. As the iterations proceed, swaps between pairs of thresholds are considered and, if the merit function improves, the swap is accepted. Because no individual gray level patterns are generated, there is no need to reconstruct the array at the end.

By comparing thresholds, all levels are optimized simultaneously. This enables a more uniform optimization to be done over the whole gray-level range of the halftone dot, and it avoids the problem of optimizing some levels so well that there is no room to optimize the last few levels.

The extension of stochastic screens to color involves the same issues that were discussed with vector error diffusion. The same problem of combining three different colored texture patterns without introducing new low-spatial-frequency texture patterns, or any new color errors, exists for stochastic screens. The solution is the same — to generate the arrays simultaneously with an optimization criteria designed to minimize low-spatial-frequency patterns. Techniques have been developed[59,62] to generate color stochastic screens with varying degrees of success.

The stochastic screen offers an interesting alternative to error diffusion. Because it is a threshold array, it is easy to compute the output halftone image. Error diffusion, especially in color, is a more complex algorithm and requires much more computation.

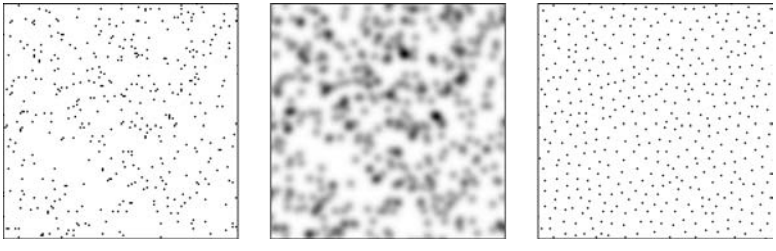


Figure 6.57 Stochastic screen seed, errors, and pattern for level 249.

This simplicity, however, comes at a cost. The constraint, required of a threshold array, that a pixel set at one gray level cannot be unset at a different gray level, puts limits on the type of patterns that can be generated at all gray levels. Because it is an adaptive algorithm, error diffusion does not have this restriction; therefore, it has the potential to generate much more visually pleasing patterns at all gray levels and colors.

6.9 Calibration

6.9.1 Introduction

Most current hard-copy devices, including inkjet printers and laser printers, are essentially binary devices; i.e., the output dots by such devices are either on or off. To use such devices to represent a continuous-tone image, the image has to be halftoned first. We may describe a such halftone printer by the diagram shown in Figure 6.58. A continuous-tone image, usually represented by eight-bit signals, is processed by a digital halftoning unit and converted to a binary image, represented by one-bit signals. The binary image is further sent to the physical output device to generate a hard copy. For color printing, the typical representations of the continuous-tone image and the binary image are eight-bit and one-bit CMYK signals, respectively. By considering the halftoning process and the binary printer together as a complete unit, we may treat the halftone printer as a continuous-tone printer, which takes eight-bit input signals and generates a hard copy. Therefore, various continuous-tone printer models, used for printer characterization, calibration, and other applications,[63] can be applied to halftone printers as well. On the other hand, a true binary printer model should be halftoning independent and should predict the output color at the pixel level from a binary input image. There will be two main advantages to having the true binary printer model. First, it will make the printer calibration halftone-algorithm independent. So, it will be possible to use different halftone methods or halftone screens without repeated and often time-consuming calibration for each halftone choice. Second, it will provide prediction of the color output at the pixel level, while all continuous-tone printer models can only

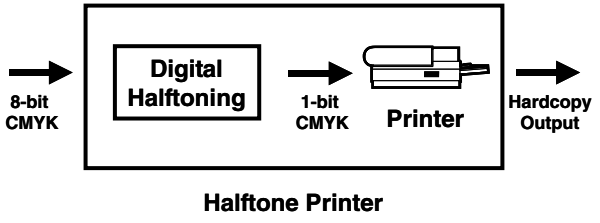


Figure 6.58 Digital halftone printer consisting of a halftoning unit and a physical binary printer.

estimate spatially averaged colors for halftone printers. Thus, using a binary printer model will allow one to preview the output image without actual printing and to compare the predicted output with the desired one in a pixel-by-pixel manner. Further image processing might follow the comparison for color correction, image enhancement, and many other possible applications. Up to now, very few binary printer models have ever been proposed, and most of the study has been limited to black-and-white (monochromatic) binary printers.[64–67] In this section, we will describe a binary printer model based on a novel two-by-two (2×2) centering concept. First proposed by Wang, it can be applied to black-and-white printers[68,69] as well as color printers.[70,71] The 2×2 printer model provides a truly measurement-based method to characterize a variety of binary printers, regardless of the actual physics and mechanics behind printing.

6.9.2 *Dot overlapping*

An idealized color printer is expected to print all dots spatially uniformly, in perfect square shape, and to have no overlapping between adjacent dots. For example, the dot pattern shown in Figure 6.59 consists of four different color pixels that are perfect cyan, green, white, and black squares, generated by an idealized CMY color printer. Because all four colors above are among the eight primaries, or solid colors, used by a CMY printer, the color appearance of each pixel can be exactly predicted from measurement of patches printed in corresponding solid colors. Because there is no overlapping between adjacent dots, no color other than primary colors needs to be considered. The idealized printer model can describe the color output at the pixel level for any binary input patterns.

In reality, the actual outputs from different printers are much more complicated than the idealized case. Dots have irregular shapes, and the size, shape, and density vary with time and location. The scattering of the light in the paper substrate adds further complexity, making a detailed

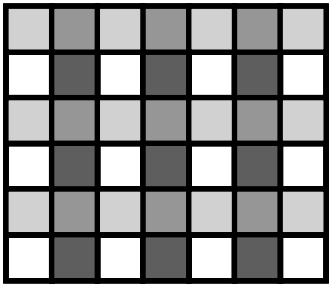


Figure 6.59 ***(See color insert)*** Idealized printer model with all output pixels in perfect square shape. The input is a spatially and periodically repeated 2×2 binary pattern with cyan, yellow, green, and black, four different colors.

microscopic model of the overlapping very difficult to construct. A circular-dot printer model, shown in Figure 6.60, is perhaps still too idealized in fact. The circular-dot model assumes that all dots have an identical round shape and size, and each circular dot is located at the center of the square defined as the output pixel. Because, to the first order, the human visual response can be assumed to be a lowpass filter, it is not necessary to have a complete knowledge of the overlapping for printer modeling. Instead, an estimation of the average color appearance of each output pixel will be sufficient. Clearly, with overlapping considered, the average color of each square pixel depends not only on the dot centered at this pixel but also on the surrounding dots — at least eight immediate neighbors. For example, in the pattern shown by Figure 6.60, each cyan dot is surrounded by two green dots, two white dots (spaces), and four black dots. Because each dot has eight possible primary colors for a CMY color printer, the total number of all possible combinations of 9 dots, including the center one and eight immediate neighbors, is given by the power 8^9, or 134,317,728. This number is definitely too large to handle for any possible characterization attempts. Furthermore, even a simple dot pattern such as the one shown in Figure 6.60 is a combination of four different 3×3 dot combinations, which are centered by cyan, green, white, and black dots, respectively. Measurement of a printed patch with the above dot pattern will only provide a result of mixed colors by four different output pixels. Finding out the contribution from each type of combinations requires additional calculation.

Pappas and others studied printer modeling for monochromatic binary printers,[65] which have only two primary colors: black and white. So, the total number of all possible 3×3 dot combinations is equal to 2^9, or 512. Assuming that the overlapping is symmetric about both x and y directions, one can reduce the number 512 to 102. This number can be further reduced to 50 under specified assumptions.[65] In other words, there are 50 different reflectance values to be determined for all possible output pixels from a black-and-white printer. Unfortunately, even with this much reduced number, it is still not trivial to solve for individual reflectance. The characterization

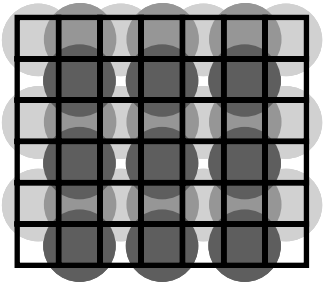


Figure 6.60 ***(See color insert)*** Circular-dot printer model showing geometrical overlap of adjacent dots from an input with the same binary pattern as the one in Figure 6.59.

process conducted by Pappas and others might be briefly described as follows: 200 different patches with periodically repeating binary structures were generated and printed by a specific printer, and the average reflectance from each patch was measured. The result of each measurement was considered as a linear combination of 50 unknown parameters, and the determination of these parameters became a matrix inversion problem. Because the matrix involved was not with a full rank, and there were discrepancies in the measurement, the solution had to be solved through a rather complicated optimization calculation. Practically, it is currently unfeasible to extend this approach to characterizing any color printers.

6.9.3 Two-by-two centering concept

Before the discussion of the 2×2 centering concept, it should be worthwhile to remark that a solid color, which has all pixels printed with exactly the same color, can be determined by a macroscopic measurement — not just for the average color of a large area but also for each individual pixels. A true continuous-tone printer can generate solid colors for any color values. Therefore, it is not difficult to characterize a continuous-tone color printer using direct measurement-based methods. However, a halftone color printer can print only a few primary colors (e.g., cyan, magenta, yellow, red, green, blue, black, and white for a CMY color printer) as solid colors. No binary patterns, other than primaries, can be considered as solid colors at the pixel level.

The difference between the 2×2 centering concept and the traditional approach is in the definition of output pixels. There is no physical reality directly tied with the square-shape output pixels. The grid defining the output pixels is a conceptual coordinate for modeling purpose. Any change of the grid, or the definition of output pixels, will not affect the actual physical output of the printer at all. Therefore, we may redefine the output pixels by shifting the grid to a new position, as shown in Figure 6.61, so that each dot, representing the physical output, will be centered at one cross point

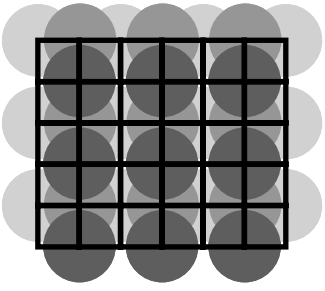


Figure 6.61 ***(See color insert)*** Same dot-overlapping pattern in Figure 6.60 is shown with the 2×2 centering coordinate.

of the shifted grid. Although the dot pattern in Figure 6.61 is exactly the same one in Figure 6.60, overlapping details within output pixels defined by the new grid are completely different from output pixels defined by a conventional coordinate, shown in Figure 6.60. In Figure 6.61, there are four different output pixels, which are also displayed separately in Figure 6.62. It is not difficult to see that all these four patterns are mirror images of each other. Hence, all pixels in Figure 6.62, as well as all pixels in Figure 6.61, should have an exactly same averaged color over the pixel area. In other words, the binary pattern of Figure 6.61 may be considered as a "solid" color. By looking at another dot pattern, shown in Figure 6.63, with a more-or-less arbitrary binary input, one may find that the output pixel, indicated by a heavy outline square, has exactly the same "solid" color defined by the 2×2 pattern in Figure 6.61. Obviously, the rest of the output pixels in Figure 6.63 also have their corresponding "solid" colors, respectively. As a matter of fact, all output pixels defined by the 2×2 centering concept are "solid" colors and can be determined by a straight measurement-based characterization process, including generating patches with periodically repeated $2 \times$

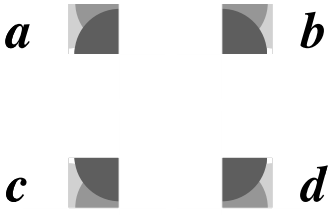


Figure 6.62 ***(See color insert)*** Four different output pixels in Figure 4 are illustrated as mirror images of each other.

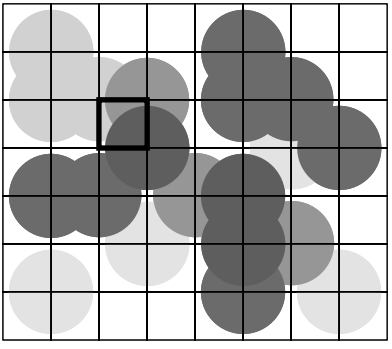


Figure 6.63 ***(See color insert)*** Dot overlap with an arbitrary binary input is shown with the 2×2 centering coordinate. The output pixel indicated by a heavy outline has the "solid" color, or the 2×2 color, defined by the binary pattern of Figure 6.59.

2 binary patterns and measuring patches macroscopically by a conventional color measurement instrument.

Because each 2×2 pattern is defined by four color input values, the total number of different 2×2 patterns, or 2×2 colors, depends on the total number of different input colors, or primary colors, of the color printer. One may prove that the total number of 2×2 colors, N, for a color printer with P primary colors, is given by the following equation:

$$N = 6C_P^4 + 9C_P^3 + 5C_P^2 + C_P^1 \tag{6.20}$$

where

$$C_P^k = \frac{P!}{(P-k)! \cdot k!} \qquad \text{if } k \le P$$

$$= 0 \qquad \text{otherwise}$$

As listed in Table 6.4, a black-and-white printer has $N = 7$ 2×2 colors, a CMY color printer has $N = 1072$, and a CMYK color printer has $N = 16{,}576$. Figures 6.64 and 6.65 show all calibration patches used for characterizing 2×2 colors for a black-and-white printer and a CMY color printer, respectively.

Table 6.4 Total Number, N, of Different 2×2 Colors for a Binary Color Printer with C Different Colorants and P Different Primary Colors

No. of Colorants, C	No. of Primary Colors, $P = 2^C$	Power P^4	No. of Independent 2×2 Colors
1	2	16	7
2	4	256	76
3	8	4,096	1,072
4	16	65,536	16,576

Clearly, the 2×2 printer model can be also applied to printers with dot shapes other than circles and with different resolutions between x and y directions. In general, the 2×2 printer modeling requires only that all overlapping dots should be symmetric about both the x and y axes and should be no larger than the size of two output pixels in both dimensions. Details of the analysis and generalizations of the 2×2 centering concept can be found in a recent paper by Crounse.[72] These conditions are well satisfied by many binary output devices. Even if the above conditions do not hold exactly, the 2×2 modeling still provides the best estimation of the output colors at the pixel level — at least in the statistical sense.

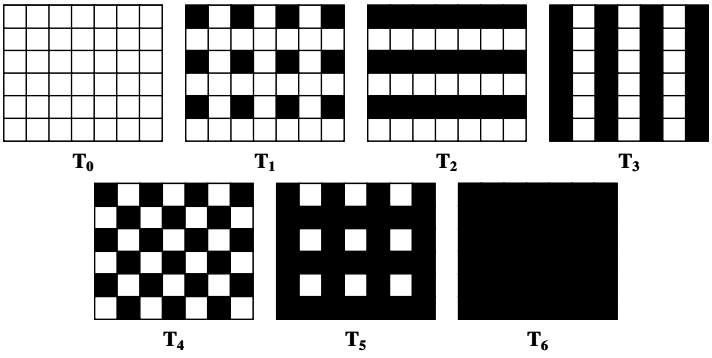


Figure 6.64 Seven 2 × 2 patterns defining seven "solid" gray levels, or 2 × 2 colors, generated by a black-and-white printer.

6.9.4 *Neugebauer equations and Yule–Nielsen modification*

The 2 × 2 printer model predicts output colors at the pixel level. To estimate the average color of a large area by a binary color printer, one may use the modified Neugebauer equations described in this section.

For binary printers, the Neugebauer equation[73] predicts average colors through combination of the primary colors. For the following discussion, colors are specified in spectral reflectance, although they could be in tristimulus values XYZ as well. The predicted color reflectance $R(\lambda)$ of an output by a binary printer is given by

$$R(\lambda) = \sum_{i=1}^{N} a_i R_i(\lambda) \tag{6.21}$$

where a_i and $R_i(\lambda)$ are the area coverage and the spectral-reflectance of each primary color, respectively, and N is the total number of primary colors. For black-and-white printers, $N = 2$. For CMY three-color printers, $N = 8$.

Considering the scattering of light within the paper, Yule and Nielsen modified the Neugebauer equation by the following equation:[74]

$$R(\lambda)^{1/n} = \sum_{i=1}^{N} a_i R_i(\lambda)^{1/n} \tag{6.22}$$

where n is the Yule–Nielsen factor, often chosen as a fitting parameter.

The difficulty in applications of Neugebauer equations with conventional printer models is to accurately estimate the area coverage of primaries, mainly due to the complexity of dot overlapping. However, this difficulty should be completely overcome by the 2 × 2 printer model.

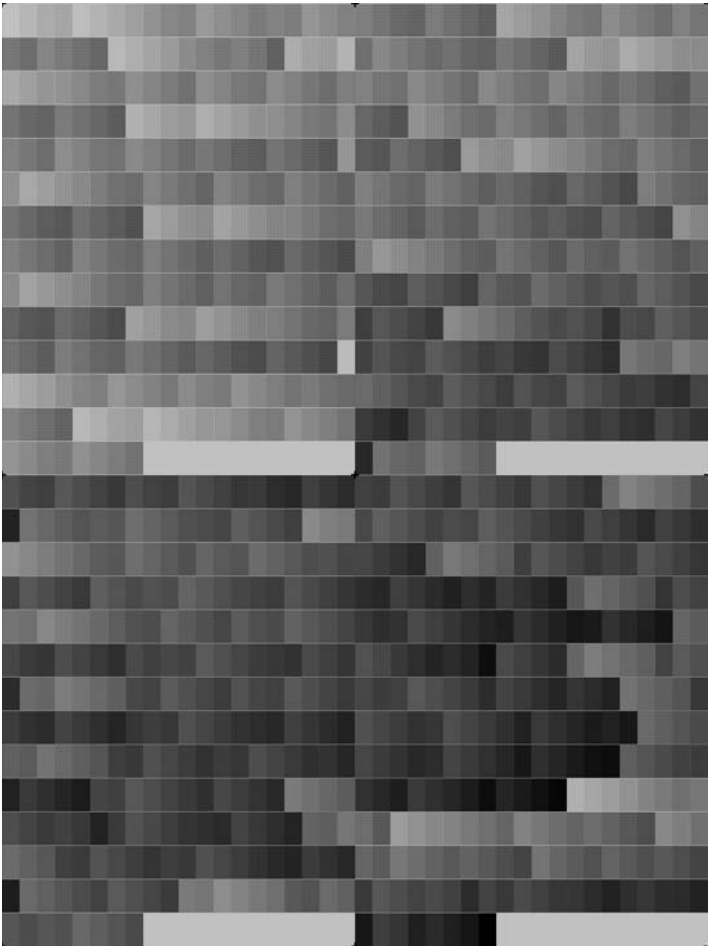


Figure 6.65 ***(See color insert)*** A page consisting of 1072 2 × 2 color patches used to characterize a CMY color printer.

The modified Neugebauer equation, shown by Equation 6.22, can be directly applied to the 2 × 2 printer model for predicting average colors of any dot combinations. As the case stands, $R_i(\lambda)$ in Equation 6.22 represents the measured spectral reflectance of each 2 × 2 color. The area coverage a_i of each 2 × 2 color is directly proportional to its occurrence m_i in number of pixels and can be calculated by the following equation, where N is the total number of different 2 × 2 colors involved:

$$a_i = \frac{m_i}{\sum_{j=1}^{N} m_j} \tag{6.23}$$

Again, although the spectral reflectance $R(\lambda)$ is used to describe the Neugebauer equations above, tristimulus values XYZ or other color measures might be also suitable for most applications. By storing all measured 2×2 colors in the power of $1/n$ into a lookup table, estimation of the average color of any binary input pattern by a 2×2 printer model should be pretty straightforward.

6.9.5 *Calibrating 2×2 printer models with reduced measurement*

The 2×2 printer model is truly measurement based, i.e., all 2×2 colors can be directly measured macroscopically. The accuracy of the 2×2 printer model can be quite high, as long as the physical dot size really matches the claimed printer resolution. Because the industry has developed several automated color measurement instruments (e.g., the Spectrolino spectrometer by Gretag and the DTP-41 scanning spectrophotometer by X-Rite), measuring all 2×2 colors, even for a four-colorant CMYK printer, is not impractical. However, for some applications, it would be appreciated to have a smaller measurement base. Due to the intrinsic geometry of the 2×2 patterns, all 2×2 colors can be quite accurately predicted by measuring and rendering only a small part of the complete 2×2 set. In general, for three-colorant (CMY) and four-colorant (CMYK) printers, most 2×2 colors made of three or four colorants can be easily predicted based on the measurement of all 2×2 colors made of one or two colorants. For example, the 2×2 color shown in Figure 6.61, as shown by the 2×2 pattern on the top of Figures 6.66 and 6.67, is made of three colorants (cyan, magenta, and yellow) by a CMY printer. It is possible to decompose this 2×2 pattern into three 2×2 patterns made of single colorant, as shown in Figure 6.66, or into three 2×2 patterns made of two colorants, as shown in Figure 6.67. There are many possible approaches that can provide satisfactory results for calibrating the 2×2 printer model with reduced measurement base. Interested readers can find the details of one method reported by Wang, Fan, and Quan.[71] Table 6.5 shows the choices of sub-sets used for the reduced measurement with this method. The number of measured color patches has been reduced from 1072 to 224 for characterizing a CMY printer, and from 16,576 to 468 for a CMYK printer.

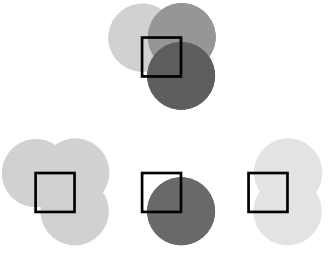


Figure 6.66 ***(See color insert)*** Decomposition of a 2×2 color made of three colorants into three 2×2 colors made of a single colorant.

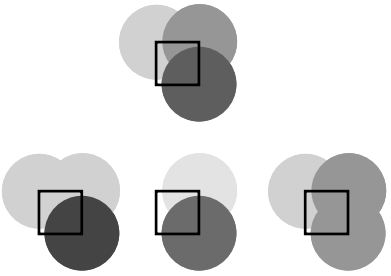


Figure 6.67 ***(See color insert)*** Decomposition of a 2×2 color made of three colorants into three 2×2 colors made of two colorants.

Table 6.5 Two-by-Two Patterns Used for Characterizing CMY or CMYK Printers with Reduced Measurement Bases

Type of 2×2 Colors	CMY Printers	CMYK Printers
White	1	1
One colorant	18	24
Two colorant	189	378
Additional with primaries	16	65
Total used for characterization	224	468
Total number of 2×2 colors	1072	16,576

6.9.6 *Halftone printer characterization*

An immediate application of the 2×2 printer model is in characterizing halftone printers. Two levels of characterization can be implemented using 2×2 printer models: for the individual channels, and for the full color characterization. Of course, characterizing a single channel of a color printer is essentially the same as characterizing a black-and-white (monochromatic) halftone printer. For a given halftone method, the halftone output with any constant input is a determined binary pattern. Using the 2×2 printer model and the measurement of all 2×2 colors, we can quickly calculate the expected average color of the determined binary pattern using the modified Yule–Nielson equation, Equation 6.22.

At the first level of characterization, fundamentally only seven patches, shown by Figure 6.64, need to be generated and measured. To determine an optimized Yule–Nielson correction, i.e., to optimize the coefficient n in Equation 6.22, additional patches with known binary patterns should also be printed and measured at the same time as for the seven 2×2 colors, or 2×2 gray levels. For a chosen halftone screen or a halftone method, a tone reproduction curve (TRC), which would be obtained traditionally by measuring patches with sampled input levels and certain data-fitting techniques, can be

fully calculated based on the measured seven 2×2 gray levels. Typically described as halftone calibration, an inversion procedure is usually applied to the characterization result for correcting any nonlinear effect caused mainly by dot overlapping. A method to apply the 2×2 printer model to calibrate the error-diffusion halftoning method is described in Reference 68. Figure 6.68 shows the comparison of the output of an uncalibrated error-diffusion method and results of calibration by the 2×2 printer model with the Yule–Nielson coefficient $n = 1$ and $n = 3$, respectively.

As the full-color level characterization, a 2×2 color printer model provides a complete description of the specified binary printer with any chosen color halftone algorithm and/or halftone screens. For example, with the measurement of all patches shown in Figure 6.65 for characterizing a CMY binary printer, we can calculate output colors for any chosen halftone screens equipped to the specific printer with any CMY input values. An experiment was conducted for characterizing a CMYK binary printer with reduced measurement. As reported also in Reference 70, a total of 488 patches representing the selected subset of 2×2 colors, listed in Table 6.5 for CMYK printer characterization, were printed on one page at 600×600 dpi resolution from a color xerographic printer and measured in spectral reflectance. The complete set of independent 2×2 colors, 16,576 in total for a CMYK printer, was derived by a rendering method described in the Reference 70. A separate page with 500 patches representing different CMYK input values was gen-

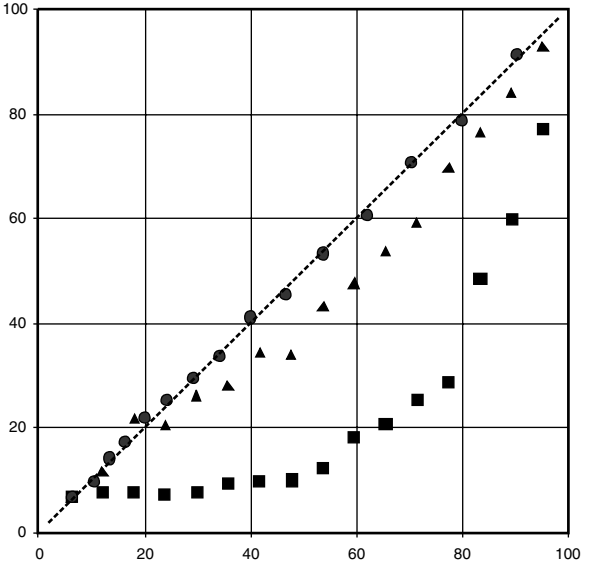


Figure 6.68 Measured output reflectance vs. desired input: (a) ■, uncalibrated error diffusion method, (b) ▲, calibrated by the 2×2 printer model with Yule-Nielson coefficient $n = 1$, and (c) ●, calibrated by the 2×2 printer model with $n = 3$.

erated by a selected halftone screen set, which had four different rotated cluster halftone screens for CMYK, respectively, in the frequency range about 150 lpi. Using the 2×2 printer model and the derived 2×2 colors, the expected colors of the 500 halftone patterns were calculated and compared with the measurement. The average ΔE between the calculation and the measurement is about 3.7, and the maximum ΔE is about 8.3.

6.9.7 *Feedback using a 2×2 printer model*

Using feedback with 2×2 printer models for image enhancement was proposed by Wang and Fan.[75,76] As an example, in this section, we briefly describe a system for eliminating *subject moiré* caused by rescreening. The topic has also been addressed for black-and-white images in other publications.[77,78] When a halftone image is copied, different artifacts might be introduced during the copying process. A subject moiré is referred to the interference between the original halftone screen(s) of the image and the halftone screen(s) used for printing the copy.

Because color printing is a nonlinear color mixing process, a cross-channel effect, such as the two-color moiré, may not show up as a low-frequency artifact in individual channels but may be present on the actual printout. Thus, it is necessary to compare the actual printout and the original image in a common color space for evaluating image quality and detecting possible artifacts. Usually, this comparison is conducted manually by the operator with an original image and an actual printed output. But it is also possible to apply a 2×2 printer model to predict the output appearance from a processed CMYK image without the actual printing process being performed. In other words, pixel by pixel, the 2×2 printer model maps the binary CMYK image to an output image represented in the same color space as the original input image. Therefore, a human vision model, which might be a simple lowpass filtering process, can be applied to emulate an operator examining the physical printout for detecting moiré or other artifacts. Furthermore, prediction of the potential artifacts from halftoning and printing allows us to make corresponding adjustments to the process to reduce or eliminate the actual occurrence of any artifact.

The flow chart in Figure 6.69 illustrates the proposed system for moiré-free color halftoning. The process starts with an initial input image specified in a standard color space (say, the CIE Lab space). Using the result of printer calibration, the CIE Lab image is converted into a continuous-tone CMYK image. Next, the continuous-tone image is converted into a binary CMYK image by the halftone process. Instead of sending the binary image to drive the printer for output, a calibrated 2×2 printer model is applied to convert the binary CMYK image to a CIE Lab image, which represents the predicted appearance of the potential printout from the chosen CMYK printer. As the ultimate goal of printing is to reproduce the original image, the entire process, combining three boxes in Figure 6.69 (printer calibration, color halftoning and printer modeling), is theoretically equivalent to an identity system.

Although each individual process (especially halftoning and printer modeling) has been known as a nonlinear process, the combined process here is really close to a linear system in the low-frequency range, at least for a well-calibrated and conducted halftone system. Therefore, we may adapt the well known feedback concept for system control to the halftone process. Because an identity system has a unity gain to compensate any deviation of the output from the input, we need a negative feedback of the exact amount of the deviation. Based on the analysis above, a feedback path, as shown in Figure 6.69, is added to the conventional halftone system. The difference between the estimation by the 2×2 printer model and the initial input is calculated and further processed by a lowpass filter. From the previous discussion, it is clear that the linear system analogy is true only for the image content in the low-frequency range. The lowpass filter should be close to the human vision model and also can be optimized based on the halftone method used in the system. One may interpret the filtered color difference as a detected moiré or artifact due to the halftoning and/or printing.

To achieve moiré-free halftoning, we subtract the detected color difference from the original input image to yield an adjusted CIE Lab input image. This adjusted input is then converted by the printer calibration to a continuous-tone CMYK image and further processed by halftoning again. The halftone result can then be sent to the CMYK printer for a hard copy as the final output of the proposed moiré-free halftoning system. Obviously, the feedback process may be conducted with more than a single feedback loop for improved moiré removal.

In the discussion above about the moiré-free halftoning, we used the CIE Lab space as a standard color space for defining the input image as well as for the 2×2 printer modeling. However, it also should be noticed that the proposed feedback process can be applied to any other measurable color spaces, such the XYZ color space, or the reflectance or density for grayscale images. The linear-system analogy used in our previous analysis is still true for any other color space, as long as all corresponding color calibrations in the system are conducted properly for the specific space.

To demonstrate the feedback technique for moiré-free halftoning, a picture of the chapel of University of Rochester was scanned from a previously generated halftone image with unknown halftone screens. The scanned

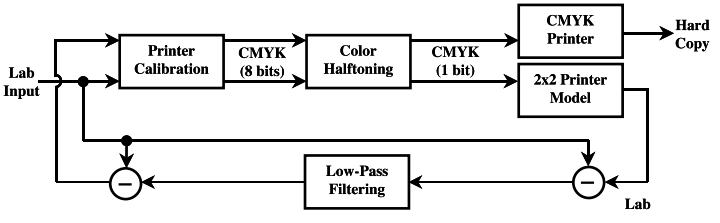


Figure 6.69 Moiré-free color halftoning system.

image was converted to the CIE Lab space by the scanner calibration and was used as the original input for the experiment described here. The input image was converted to a continuous-tone CMYK image and was then halftoned using selected halftone line screens. The CMYK binary output of halftoning is shown in Figure 6.70a. One may see that a moiré with a very low spatial frequency near 45 is clearly shown on the halftone picture due to the interference between the line screens and the previously generated unknown halftone structure. A 2×2 printer model was calibrated by the procedure discussed previously with a chosen CMYK printer. The application of the 2×2 printer model to the binary halftone image, shown in Figure 6.70a, provided an estimated color appearance of the projected printer output. The difference between the original input and the predicted output was calculated and further processed by a lowpass filter. The filtering was conducted in the spatial domain with a 7×7 standard lowpass filter corresponding to a cutoff frequency approximately equal to 70% of the line frequency

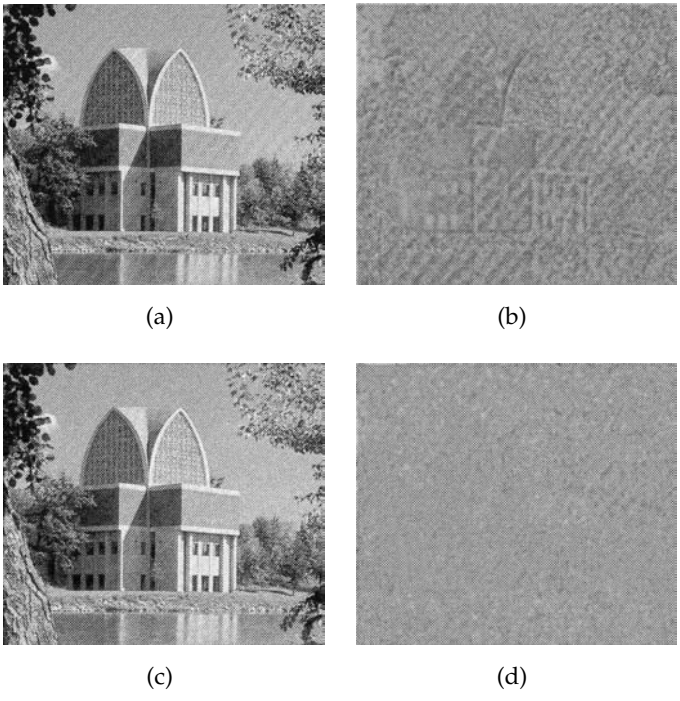


Figure 6.70 ***(See color insert)*** (a) Halftone output from a scanned image with unknown halftone structures and without feedback compensation. (b) Low-pass-filtering output (with enhanced contrast for display) from the difference between the original input image and the predicted color output of a by the 2×2 printer model. (c). Final halftone output with moiré-free compensation. (d) Low-pass-filtering output from the difference between the original input image and the predicted color output of the final halftone image c by the 2×2 printer model.

of the line screens used in this experiment. The filtered difference image is shown in Figure 6.70b with a constant bias added and contrast enhanced for the demonstration purpose. The difference image was then used as the feedback subtracted from the initial input image to generate a compensated input. The second-round halftoning created an improved halftone image, shown in Figure 6.70c. For further comparison, the difference between the original input image and the estimated color appearance by the 2×2 printer model from the final output in Figure 6.70c was also calculated. The result, shown in Figure 6.70d with the same bias and contrast as Figure 6.70b, can be compared with the first-round result in Figure 6.70b, and the improvement, in terms of moiré-removal, is quite clear.

References

1. Talbot, W. H. F., Improvements in the Art of Engraving, British Patent Specification #565, October 29, 1852.
2. Levy, L. E. and Levy, M., Screen for Photomechanical Printing, U.S. Patent No. 492, February 21, 1893, 333.
3. Hepher, M., A comparison of ruled and vignetted screens, *Penrose Annu.,* 47, 166–177, 1953.
4. Roberts, L. G., Picture coding using pseudo-random noise, *IRE Trans.on Infor. Theory,* IT-8, 145–154, 1962.
5. Judice, C. N., Jarvis, J.F., and Ninke, W.H., Using ordered dither to display continuous tone pictures on an AC plasma panel, *Proc. SID,* 15, 4 161–169, 1974.
6. Bayer, B. E., An optimum method for two-level rendition of continuous-tone pictures, *IEEE Int. Conf. Commun.,* 1, 11–15, 1973.
7. Klensch, R. J., Meyerhofer, D., and Walsh, J.J., Electronically generated halftone pictures, *TAGA Proc.,* 22, 302–320, 1970.
8. Roetling, P. G., and T. M. Holladay, Tone reproduction and screen design for pictorial electrographic printing, *J. Appl. Photogr. Eng.,* 5(4), 179–182, 1979.
9. Levien, R., Output dependent feedback in error diffusion halftoning, in *Proc. IS&T 46th Ann. Conf.*, May 15–20, Cambridge, MA 1993, 115–118.
10. Eschbach, R., Pixel-based error-diffusion algorithm for producing clustered halftone dots, *J. Electr. Imaging* 3(2), 198–202, 1994.
11. Analoui, M. and Allebach, J.P., Model-based halftoning by direct binary search, *Proc. 1992 SPIE/IS&T Symp. on Electronic Imaging Science and Technology,* 1666, February 9–14, San Jose, CA, 1992, 96–108.
12. Eschbach, R. and Hauck, R., Binarization using a two-dimensional pulse-density modulation, *JOSA A.*, 4, 1873–1878, 1987.
13. Scheermesser, T., Broja, M., and Bryngdahl, O., Adaptation of spectral constraints to electronically halftoned pictures, *JOSA A.,* 10, 412–417, 1993.
14. Rhodes, W. L. and Hains, C. M., The influence of halftone orientation on color gamut and registration sensitivity, in *Proc. IS&T 46th Annual Conference*, May 9–14, 1993, Cambridge, MA, also in Eschbach, R., *Recent Progress in Digital Halftoning,* IS&T publ. 1994, 117–119.
15. Knox, K. T., Edge enhancement in error diffusion, extended abstract, SPSE annual meeting, Boston, MA, May 1989.

16. Roetling, P. G., Analysis of detail and spurious signals in halftone images, *J. Appl. Photogr. Eng*. 3, 12, 1977.
17. Holladay, T. M., An optimum algorithm for halftone generation for displays and hard copies, *Proc. SID*, 21(2), 185–192, 1980.
18. Clapper, F. R. and Yule, J. A. C., The effect of multiple internal reflections on the densities of halftone prints on paper, *J. Opt. Soc. Am*., 43(7), 600–603, 1953.
19. Hunt, R. W. G., *The Reproduction of Colour in Photography, Printing, and Television*, 4th ed., Fountain Press, England, 1987.
20. Tollenaar, D., Moiré interference phenomena in halftone printing, in *Moiré Pattern in Printing*, Research and Engineering Council of the Graphic Arts Industry, Arlington, VA, 1960.
21. Field, G., *Color and Its Reproduction*, Graphic Arts Technical Foundation, Pittsburgh, PA, 1988, 229, 232.
22. Lamming, M. G. and Rhodes, W. L., A simple method for improved color printing of monitor images, *ACM Trans. Graphics*, 9(4), 345, 375, 1990.
23. Yule, J. A. C., *Principles of Color Reproduction,* Wiley, New York, 1967, 337–342.
24. Wurzburg, F. L., Jr., Understanding moiré screen pattern, *Gravure* 7(9), 17–21, 1961.
25. Pollak, F., The dependence of the contrast of moiré patterns on the colors of the printing inks, *Proc. Technical Association of the Graphic Arts 10th Annual Meeting,* 1958, 37–44.
26. Chen, J., An investigation of color variation as a function of register in dot-on-dot multicolor printing, *Proc. Technical Association of the Graphic Arts*, 1984, 315–334.
27. Gustavson, S., *Dot Gain in Colour Halftones*, Linköping Studies in Science and Technology Dissertation No. 492, September 1997.
28. Neugebauer, H. E. J., Comparison of the gamut of three different methods of color reproduction, Haloid Physics Department Report No. 114, 1958.
29. Delabastita, P., Screening techniques, moiré in four color printing, *Proc. TAGA Conference,* April, 44, 1992.
30. Daels, K. and Delabastita, P., Tone dependent phase modulation in conventional halftoning, *Proc. IS&T 47th Annual Conference/ICPS 1994,* 472.
31. Kang, H. R., *Digital Color Halftoning,* SPIE Optical Engineering Press, Bellingham, WA, 1999, 133.
32. McKinney, P., Screen patterns in printing, *Am. Pressman*, 66(4), 16–20, 53–58, April 1956 and *Moiré Pattern in Printing,* Research and Engineering Council of the Graphic Arts Industry, Arlington, VA, 1960.
33. Amidror, I., The moiré phenomenon in color separation, *Raster Imaging and Digital Typography II — The Proc. 2nd International Conference on Raster Imaging and Typography,* R. Morris and J. Andre, Eds., Cambridge University Press, Boston, MA, 1991, 98–119.
34. T. M. Holladay, An optimum algorithm for halftone generation for displays and hard copies, *Proc. SID,* 21, 185–192, 1980.
35. Wang, S., Fan, Z., and Wen, Z., Nonorthogonal halftone screens, *Proc. NIP18: Int. Conf. on Digital Printing Technologies,* 2002.
36. Scheermesser, T., Wyrowski, F., and Bryngdahl, O., Digital halftoning using two-dimensional carriers with a noninteger period, J. *Electr. Imaging,* 4, 40–47, 1995.
37. Fink, P., *PostScript Screening: Adobe Accurate Screens*, Adobe Press, Mountain View, CA, 1992, 43–61.

38. Keller, H., Koll, R., and Taudt, H., Method for the electro-optical reproduction of half-tone pictures, U.S. Patent 4,084,183, 1978.
39. Baqai, F. A. and Allebach, J.P., Computer-aided design of clustered-dot color screens based on a human visual system model, *Proc. IEEE,* 90–1, 104–122, 2002.
40. Floyd, R. W. and Steinberg, L., An adaptive algorithm for spatial grey scale, *Proc. Dig. SID Int. Symp.*, Los Angeles, CA, 1975, 36–37.
41. Ulichney, R. A., Dithering with blue noise, *Proc. IEEE,* 76(1), 56–79, 1988.
42. Knox, K. T., Edge enhancement in error diffusion, *Proc. SPSE 42th Ann. Conf.*, May 14–19, Boston, MA, 1989, 310–313.
43. Kim, J. H., Chung, T.I., Kim, J. S., Son, K. S., and Kim, Y. S., New edge-enhanced error diffusion algorithm based on the error sum criterion, *J. Electr. Imaging,* 4(2), 172–178, 1995.
44. Shiau J. N. and Fan, Z., A set of easily implementable coefficients in error diffusion with reduced worm artifacts, in Color imaging: device-independent color, color hardcopy, and graphic arts, *Proc. SPIE* 2658, 222–225, 1996.
45. Jarvis, J. F., C. N. Judice, and W. H. Ninke, A survey of techniques for the display of continuous tone pictures on bilevel displays, *Comput. Graphics Image Process.,* 5, 13–40, 1976.
46. Knox, K. T., Error image in error diffusion, image processing algorithms and techniques III, *Proc. SPIE,* 1657, 268–279, 1992.
47. Knox, K. T., Evolution of error diffusion, *J. Electr. Imaging* 8(4), 422–429, 1999.
48. Yule, J. A. C., *Principles of Color Reproduction*, Wiley, New York, 1967.
49. Heckbert, P. S., Color quantization for frame buffer display, *Proc. SIGGRAPH '82*, Boston, MA, July 1982, 297–307.
50. Venable, D., Stinehour, J., and Roetling, P., Selection and use of small color sets for pictorial display, *Proc. SPSE 43rd Annu. Conf.*, Rochester, NY, May 1990, 90–92.
51. Kolpatzik, B., and Bouman, C.A., Optimized universal color palette design for error diffusion, *J. Electr. Imaging,* 4(2), 131–143, 1995.
52. Klassen, R. V. and Eschbach, R., Vector error diffusion in a distorted color space, *Proc. IS&T 47th Annu. Conf.*, Rochester, NY, May 15–20, 1994, 489–491.
53. Haneishi, H., Suzuki, T., Shimoyama, N., and Miyake, Y., Color digital halftoning taking colorimetric color reproduction into account, *J. Electr. Imaging,* 5(1), 97–106, 1996.
54. Fan, Z., Boundary artifacts reduction in vector error diffusion, in Color imaging: Device-independent color, color hardcopy, and graphic arts, *Proc. SPIE,* 3648, 480–484, 1999.
55. Fan Z. and Harrington, S., Semi-vector error diffusion for color images, in Color imaging: device-independent color, color hardcopy, and graphic arts, *Proc. SPIE,* 3648, 466–469, 1999.
56. Harrington, S., Sum and difference error diffusion, U.S. Patent, No. 6,072,591, June 6, 2000.
57. Rolleston, R., Cohen, S., and R. Morrison, Halftoning with random correlated noise, *Opt. News,* 15(9), A92, 1989.
58. Yao, M. and Parker, K. J., Modified approach to the construction of a blue noise mask, *J. Electr. Imaging,* 3(1), 92–97, 1994.
59. Spaulding, K. E., Miller, R. L., and Schildkraut, J., Methods for generating blue-noise dither matrices for digital halftoning, *J. Electr. Imaging,* 6(2), 208–230, 1997.

60. Newbern, J. and Michael Bove, Jr., V., Generation of blue noise arrays by genetic algorithm, *SPIE Proc.,* 3016, 441–450, 1997.
61. Wang, S., Stochastic halftoning screening method, U.S. Patent, No. 5,673,121, September 30, 1997.
62. Wang, M. and Parker, K. J., Properties of jointly-blue noise masks and applications to color halftoning, *J. Imaging Sci. Technol.,* 44(4), 360–370, 2000.
63. Bala, R., Chapter 5, Device characterization, in this book.
64. Roetling, P. G. and Holladay, T.M., Tone reproduction and screen design for pictorial electrographic printing, *J. Appl. Phot. Eng*., 15(4), 179–182, 1979.
65. Pappas, T. N., Dong, C. K., and Nuehoff, D. L., Measurement of printer parameters for model-based halftoning, *J. Electr. Imaging*, 2, 193–204, 1993.
66. Rosenberg, C. J., Measurement-based evaluation of a printer dot model for halftone algorithm tone correction, *J. Electr. Imaging*, 2, 205–212, 1993.
67. Knox, K. T., Hains, C., and Sharma, G., Automatic calibration of halftones, in *Proc. SPIE: Human Vision and Electronic Imaging*, B. E. Rogowitz and J. P. Allebach, Eds., January 1996, 2657, 432–436.
68. Wang, S. G., Knox, K. T., and George, N., Novel Centering Method for Overlapping Correction in Halftoning, in *Proc. IS&T 47*th *Annu. Conf.*, 1994, 482–486.
69. Wang, S. G., Two-by-two centering printer model with Yule–Nielsen equation, *Proc. IS&T NIP14*, 302–305, 1998.
70. Wang, S. G., Algorithm-independent color calibration for digital halftoning, in *Proc. Fourth Color Imaging Conference*, 1996, 75–78.
71. Wang, S. G., Fan, Z., and Quan, S., Two-by-two centering printer model, *Proc. Fourth Int. Conf. on Imaging Science and Hardcopy (ICISH'01)*, 2001, 1–4.
72. Crounse, K. R., Measurement-based printer models with reduced number of parameters, *Proc. SPIE,* 4663, 121–129, 2002.
73. Neugebauer, H. E. J., Die theoretischen grundlagen des mahrjarbenbuchdrucks, *Z. Wiss. Photogr.*, 36, 73–89, 1937.
74. Yule, J. A. C. and Nielsen, W. J., The penetration of light into paper and its effect on halftone reproduction, *Proc. TAGA*, 3, 65–76, 1951.
75. Wang, S. G. and Fan, Z., Moiré-free color halftoning using 2×2 printer modeling, *Proc. SPIE*, 4300, 397–403, 2001.
76. Wang, S. G., Feedback for printer calibration, *Proc. IS&T NIP15*, 327–330, 1999.
77. Levien, R., Moiré suppression screening, in Color imaging: device-independent color, color hardcopy, and graphic arts V, *Proc. SPIE,* 4663, 121–129, 2002.
78. Roetling, P. G., Halftone method with edge enhancement and moiré suppression, *J. Opt. Soc. Amer.,* 66–10, 1985–989, 1976.

Recommended readings

Adobe Systems, Inc., *Postscript Language Reference Manual*, 2nd ed., Reading MA, Addison Wesley, 1990.

Allebach, J. P., Random nucleated halftone screening, *Photogr. Sci. Eng*., 22, 1978, 898–891.

Allebach, J. P., Visual model-based algorithms for halftoning images, *Proc. SPIE*, 310, 1981, 151–158.

Allebach, J. P. and B. Liu, Random quasi-periodic halftone process, *J. Opt. Soc. Am*., 66, 1976, 909–917.

Allebach, J. P. and B. Liu, Analysis of halftone dot profile and aliasing in the discrete binary representation on images, *J. Opt. Soc. Am.*, 67, 1977, 1146–1154.

Amidror, I., The moiré phenomenon in color separation, raster imaging and digital typography II, *Proc. 2nd International Conference on Raster Imaging and Typography,* R. Morris and J. Andre, Eds., Cambridge University Press, 98–119.

Amidror, I., R. Hersch, and V. Ostromoukhov, Spectral analysis and minimization of moiré patterns in color separation, *J. Electr. Imaging*, 3(3), July 1994, 295–317.

Anastassiou, D. and K. S. Pennington, Digital halftoning of images, *IBM J. Res. Develop.*, 26(6), 1982, 687–697.

Arney, J. S., C. D. Arney, and P. G. Engeldrum, Modeling the Yule–Nielsen halftone effect, *J. Imaging Sci. Technol.,* 40(3), May/June 1996, 233–238.

Avenel, Y., Agfa announces the end of moiré in PostScript screens, *Fr. Graphique*, 68, February 1992, 14–15.

Bartleson, C. J., Optimum tone reproduction, *J. SMPTE*, 84(8), 1975, 613–618.

Bayer, B. E., An optimum method for two level rendition of continuous-tone pictures, *Proc. IEEE Int. Conf. Commun.*, 26-11–26-15.

Billotet-Hoffman, C. and O. Bryngdahl, On the error diffusion technique for electronic halftoning, *Proc. SID*, 24, 1983, 253–258.

Blatner, D. and S. Roth., *Real World Scanning and Halftones*, Peachpit Press, Berkeley, CA, 1993.

Clapper, F. R. and J. A. C. Yule, The effect of multiple internal reflections on the densities of half-tone prints on paper, *J. Opt. Soc. Am.*, 43(7), 1953, 600–603.

Dalton, J. C., Visual Model Based Image Halftoning Using Markov Random Field Error Diffusion, thesis, University of Delaware, December 1983.

Engeldrum, P. G., The color gamut limits of halftone printing with and without the paper spread function, *J. Imaging Sci. Technol.,* 40(3), May/June 1996, 239–244.

Eschbach, R., Recent progress in digital halftoning, *Reprints of IS&T Proc.*, IS&T publication, 1994.

Eschbach, R., Recent progress in digital halftoning, II, *Reprints of IS&T Proc.*, IS&T publication 1999.

Eschbach, R. and K. Braun, Recent progress in color science, *Reprints of IS&T Proc.*, IS&T publication, 1997.

Fan, Z., Dot-to-dot error diffusion, *J. Electr. Imaging*, 2(1), January 1993, 62–66.

Fink, P., *PostScript Screening: Adobe Accurate Screens*, Mountain View, CA, Adobe Press, 1992.

Floyd, R.W. and L. Steinberg, Adaptive algorithm for spatial greyscale, *SID Int. Symp. Digest of Tech. Papers*, 1975, 36–37.

Floyd, R.W. and L. Steinberg, An adaptive algorithm for spatial greyscale, *Proc. SID*, 17(2), 1976, 75–77.

Goertzel, G., Digital halftoning on the IBM 4250 printer, *IBM J. Res. Develop.*, 31(1), January 1987.

Gustavson, S., *Dot Gain in Colour Halftones*, Linköping Studies in Science and Technology Dissertation No. 492, September 1997.

Haneishi, H., Y. Suzuki, N. Shimoyama, and Y. Miyake, Color digital halftoning taking colorimetric color reproduction into account, *J. Electr. Imaging*, 5(1), January 1996, 97–106.

Hunt, R. W. G., *The Reproduction of Colour in Photography, Printing, and Television*, 4th ed., Fountain Press, England, 1987.

Jarvis, J. F., C. N. Judice, and W. H. Ninke, A survey of techniques for the display of continuous-tone pictures on bilevel displays, *Computer Graphics and Image Processing*, 5, 1976, 13–40.

Johnson, A., Techniques for reproducing images in different media: advantages and disadvantages, *Proc. Tech. Assn. Graphic Arts*, 2, 1992, 739, 755.

Johnson, J. L., *Principles of Non Impact Printing*, Palatino Press, Irvine, CA, 1986.

Jones, P. R., Evolution of halftoning technology in the United States patent literature, *J. Electr. Imaging*, 3(3), July 1994, 257–275.

Kang, H. R., *Color Technology for Electronic Imaging Devices*, SPIE Optical Engineering Press, Bellingham, WA, 1997.

Kang, H. R., *Digital Color Halftoning*, SPIE Optical Engineering Press, Bellingham, WA, 1999.

Kermisch, D. and P. G. Roetling, Fourier spectrum of halftone images, *J. Opt. Soc. Am.*, 65, 1975, 716–723.

Knox, K. T. and R. Eschbach, Threshold modulation in error diffusion, *J. Electr. Imaging*, July 1993, 185–192.

Knox, K., T. Holladay, and R. Eschbach, Hybrid halftoning for color moiré reduction, *SID Digest of Technical Papers*, XXVI, May 1995, 837–839.

McKinney, P., Screen patterns in printing, *Am. Pressman*, 66(4), April, 1956, 16 –20, 53–58, and in *Moiré Pattern in Printing,* Research and Engineering Council of the Graphic Arts Industry, Arlington, VA, 1960.

Mitsa, T. and K. Parker, Digital halftoning technique using a blue-noise mask, *JOSA A.*, 9, November 1992, 1920–1929.

Nothman, G.A., *Nonimpact Printing*, Graphic Arts Technical Foundation, Pittsburgh, PA, 1989.

Oittinen, P., H. Utio, and H. Saarelma, Color gamut in halftone printing, *J. Imag. Sci. Technol.*, 36(5), September/October 1992, 496, 501.

Pappas, T. N., C.-K. Dong, and D. L. Neuhoff, Measurement of printer parameters for model-based halftoning, *J. Electr. Imaging*, July 1993, 193–204.

Rhodes, W. L., Proposal for an empirical approach to reproduction, *Color Res. Appl.*, 3(4), 1978, 197, 201.

Rich, D. C., Critical parameters in the measurement of the color of nonimpact printing, *J. Electr. Imaging*, July 1993, 231–236.

Roetling, P. G., Halftone method for edge enhancement and moiré suppression, *J. Opt. Soc. Am.*, 66, 1976, 985–989.

Roetling, P. G., Visual performance and image coding, *Proc. SID*, 17(2), 1976, 111–114.

Roetling, P. G., Binary approximation of continuous-tone images, *Photogr. Sci. Eng.*, 21, 60–65.

Roetling, P. G. and T. M. Holladay, Tone reproduction and screen design for pictorial electrographic printing, *J. Appl. Photogr. Eng.*, 5(4), 1979, 179–182.

Rosenberg, C. J., Measurement-based evaluation of a printer dot model for halftone algorithm tone correction, *J. Electr. Imaging*, July, 1993, 205–212.

Schreiber, W. F. and D. E. Troxel, Transformation between continuous and discrete representation of images: a perceptual approach, *IEEE Trans. PAMI*, PAMI-7(2), 1985, 178–186.

Schreiber, W. F., *Fundamentals of Electronic Imaging Systems: Some Aspects of Image Processing*, Springer-Verlag, New York, 1986.

Stoffel, J. C. and J. F. Moreland, A survey of electronic techniques for pictorial reproduction, *IEEE Tran. Commun.*, 29, 1981, 1898–1925.

Stoffel, J. C., *Graphical and Binary Image Processing and Applications*, Artech House, Boston, MA, 1982.
Stucki, P., Image processing for document reproduction, in *Advances in Digital Image Processing*, Plenum Press, New York, 177–218.
Stucki, P., MECCA — A multiple-error correcting computation algorithm for bilevel image hardcopy reproduction, *Research Report RZ1060*, IBM Research Laboratory, Zurich, Switzerland, 1981.
Stucki, P., Digital screening of continuous tone image data for bi-level rendition, in *Advances in Digital Image Processing*, Peter Stucki, Ed., IBM Zurich Research Library, New York/London, Plenum Press, 1978, 209–218.
Sugiura, S. and T. Makita, An improved multilevel error diffusion method, *J. Imaging Sci. Technol.*, 39(6), November/December 1995, 495–501.
Ulichney, R., *Digital Halftoning*, Cambridge, MA, MIT Press, 1987.
Ulichney, R., The void-and-cluster method for dither array generation, *Proc. SPIE*, 1913, February 1993, 332–343.
Wurzburg, F. L., Jr., Understanding moiré screen pattern, *Gravure*, 7(9), September 1961, 17–21, and in *Moiré Pattern in Printing*, Research and Engineering Council of the Graphic Arts Industry, Arlington, VA, 1960.
Yao, M. and K. Parker, Modified approach to the construction of a blue noise mask, *JEI, 3*, January 1994, 92–97.
Xie, Z. and M. Rodriquez, A bandwidth preservation approach to stochastic screening, *Proc. IS&T Third Technical Symp. on Prepress, Proofing and Printing*, October 31, 1993, Chicago, IL.

chapter seven

Human visual model-based color halftoning

A. Ufuk Agar
Hewlett-Packard Laboratories
Farhan A. Baqai
Sony Electronics
Jan P. Allebach
Purdue University

Contents

7.1 Introduction

Digital color halftoning is the process of transforming continuous-tone color images into images with a limited number of colors. The importance of this process arises from the fact that many color imaging systems contain output devices, such as color printers and low-bit-depth displays, that are bilevel or multilevel with only a few levels. The goal is to create the perception of a continuous-tone color image using the limited color discrimination capability and the lowpass characteristics of the spatial sensitivity of the human visual system (HVS).

The algorithms for digital monochrome halftoning can be categorized into three classes, in decreasing order of how myopically they transform a given image into a halftone and, thus, in increasing order of computational complexity and halftone quality.

In the first class are the point-wise approaches such as screening or dithering, whereby the image is compared pixel by pixel to an array of thresholds, and the pixels exceeding this spatially varying mask of thresholds are changed to black. Halftone screens are of two types: clustered-dot and dispersed-dot. Clustered-dot screens produce binary textures in which the individual printer dots are grouped into clusters. Dispersed dot screens spread the individual printer dots as far apart as possible. They work well for printing processes that produce consistently positioned and sized dots with little or no dot gain. Clustered-dot halftones render well on printers that have difficulty generating an isolated dot (e.g., electro-photographic printers). They are quite robust to dot gain and variations in the process parameters.

The next class of approaches uses the information about a neighborhood of pixels to decide the halftone state of a given pixel. Error diffusion is the most important example of this category.

The third class consists of iterative approaches whereby several passes over the image pixels are made to minimize an error metric or satisfy certain constraints before the halftoning process is completed.

All three of these algorithm classes can be generalized to digital color halftoning with some modifications, as color halftoning can be viewed as an L-dimensional generalization of the monochrome process where L depends on the number of primaries that the rendering device possesses. For example, L is equal to three for CMY (cyan, magenta, yellow) printers and four for CMYK (cyan, magenta, yellow, black) printers.

The naive digital color halftoning approach is to apply these monochrome halftoning techniques scalarly and independently to these color (RGB) or colorant planes (CMYK). As expected, this scalar approach leads to color artifacts and poor color rendition, because it does not exploit the correlation between color or colorant planes, which is a key element in our color perception and appreciation of the halftone quality.

Color printing presents many problems that we also encounter in monochrome printing. However, it also presents some problems that are unique to color printing. For instance, if the colorants are printed on top of each other, this will result in dots that contrast undesirably with the paper in the highlights. Another problem unique to color halftoning is that of *misregistration*. This is due to the difficulty of controlling the relative offsets of the different colorant planes. If the colorants are ideal, the visual appearance is independent of small registration errors.[1] Unfortunately, real colorants are not ideal, and registration errors can cause significant color shifts. Therefore, an important objective of the color halftoning scheme is robustness to registration errors and minimization of dot-on-dot printing.

As mentioned earlier, the HVS is responsible for mapping the physical color halftone image to the percept of a continuous-tone color image. To achieve this, all halftoning algorithms depend on the limited spatial resolution of the HVS. In addition, all color halftoning algorithms rely on colorimetry to assure correct macroscopic color reproduction. It is reasonable to suppose, however, that even better image quality might be achieved if the color halftoning algorithm were to make explicit use of a detailed spatiochromatic model for the HVS rather than simply depending on very general spatial properties of the HVS and using only a macroscopic color model-based on colorimetry. Indeed, this conjecture is the subject of this chapter, which deals with color halftoning algorithms that have embedded within them a detailed spatiochromatic model for the HVS.

Between the digital color halftone image and the HVS lies the physical printing mechanism, including the colorant and media. To effectively use an HVS within a color halftoning algorithm, it is essential to also take this part of the imaging pipeline into account. Thus, we discuss color hardcopy models in Section 7.2. Then, in Section 7.3, we review color human visual system models. Both topics are treated in greater depth elsewhere in this book.

Iterative search-based halftoning techniques have the flexibility and power to make the most effective use of detailed color hardcopy and HVS models. This is the subject of Section 7.4. These algorithms generate dispersed dot textures that, as mentioned earlier, are suitable for printing devices that can stably render isolated dots. Inkjet printers are perhaps the best example of such devices. Iterative search-based algorithms are generally not practical for direct use in printing applications. These methods can, however, be used as a gold standard for what is possible with dispersed-dot halftoning. In addition, they can serve as the basis for design of other halftoning algorithms that are practically implementable, such as halftone screens.

Beyond screening, error diffusion is the other dispersed-dot halftoning method that is widely used in applications. In Section 7.5, we briefly review the use of color HVS models within the context of error diffusion.

So far, we have discussed the use of color hardcopy and HVS models only with dispersed-dot halftoning algorithms. In Section 7.6, we turn our attention to the use of these models to design screens that produce clustered-dot periodic textures. Such screens largely form the basis for commercial printing and are also widely used with electrophotographic (laser) printers that are too unstable for dispersed-dot textures. Finally, in Section 7.7, we wrap up with a summary and conclusions.

7.2 Color hardcopy models

It is common practice to conceptualize a printer as producing dots that are perfect $1/R \times 1/R$-in^2 squares (with a rectangular dot profile) where R is the printer resolution in dots per inch (dpi) as shown in Figure 7.1a. Furthermore, we usually assume that the colorants are ideal, i.e., that they have the following properties that lead to the additivity of dot interactions:

1. They are transparent (do not scatter incident light).
2. Their absorptance coefficients are proportional to their concentration (Beer–Bouguer law).[2]
3. They satisfy the block dye assumption (absorptance functions have nonoverlapping rectangular shape).

The visual appearance of halftones created with ideal colorants is independent of small registration errors.[1]

In reality, most printers fail to satisfy these assumptions. They produce dots that may overlap (as depicted in Figure 7.1b), the colorants are not ideal, the colorant interactions are not additive, and the printer dot profiles for different colorants may be considerably different, as can be observed in a high-resolution scan of a color halftone.

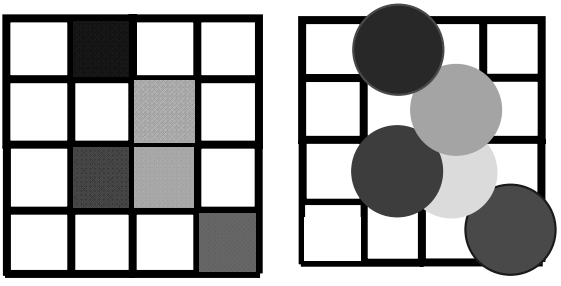


Figure 7.1 ***(See color insert following page 430)*** (a) Ideal and (b) actual printer dots.

Color halftoning algorithms that do not account for the nonideal characteristics of colorants and the complicated dot interactions suffer from incorrect color reproduction unless other corrections are introduced. Researchers have developed several techniques to model and characterize color printing, and the techniques are utilized in the design of color halftoning algorithms. Such techniques can be broadly classified into three groups. In the first group are the spectral and colorimetric printer model-based approaches that characterize the spectral distribution or the color created by a certain combination of colorants dots at a given printer-addressable location using a model of the printing process. In the second group are the dot-model-based approaches in which a model accounting for nonideal dot profiles and interactions between neighboring dots is incorporated into the halftoning algorithm. In the third group are the empirical tone-correction-based approaches that precompensate for dot interactions by modifying the image data using a lookup table (LUT) prior to halftoning and printing.

The most important spectral (colorimetric) model used in color printing is the spectral (colorimetric) Neugebauer color mixing model.[3] The spectral Neugebauer color mixing model states that $R(\lambda)$, the spectral distribution as a function of wavelength λ of a color halftone obtained using three colorants, can be written as an area-weighted sum of the reflectances of the one, two, or three color overprints of the three colorants and of the paper substrate, i.e.,

$$R(\lambda) = \sum_{i=1}^{8} w_i R_i(\lambda) \tag{7.1}$$

These eight colorant combinations are called the *Neugebauer primaries.* The term $R_i(\lambda)$ is the spectral distribution of the patch with only the ith Neugebauer primary on it, and the weight w_i is the fractional proportion of the ith Neugebauer primary in the given halftone. The monochrome version of this model is called the Murray–Davies model. The spectral distributions, measured by a Gretag–Macbeth† SPM 50 spectrophotometer, of the eight Neugebauer primaries printed with an HP DeskJet 970C three-colorant inkjet printer are shown in Figure 7.2.

The linear Neugebauer (Murray–Davies) color mixing model fails to account for an important phenomenon called *optical dot gain*. This phenomenon, first studied by Yule and Nielsen,[4] is defined as the change in the measured reflectance due to surface and subsurface scattering in the paper substrate, between and inside colorant layers. Yule and Nielsen proposed an empirical power-law correction to the Murray–Davies model. Viggiano[5] extended this correction to color and obtained the following expression:

$$R(\lambda)^{1/n} = \sum_{i=1}^{8} w_i R_i(\lambda)^{1/n} \tag{7.2}$$

† Gretag-Macbeth AG, Althardstrasse 70, 8105 Regensdorf, Switzerland.

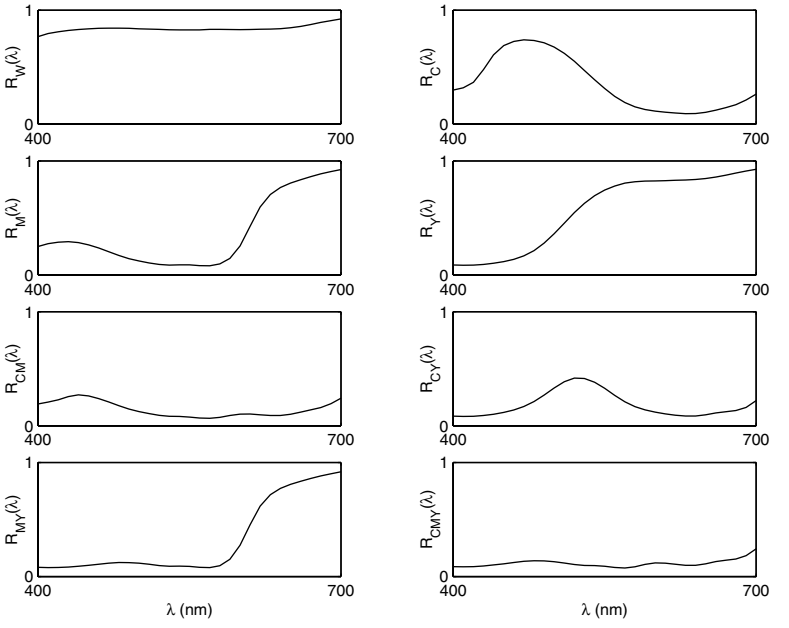


Figure 7.2 Spectra of the eight Neugebauer primaries measured from prints made with an HP Deskjet 970C printer.

where the factor n is called the Yule–Nielsen factor, and it is empirically derived from the best fit of the model to the training data set. Numerous versions of the Yule–Nielsen corrected Neugebauer models have been used to predict the spectral distributions and tristimulus values of color and monochrome halftones.[6–14] We will also use a similar model in Section 7.6.3. Other color macroscopic (at the color halftone patch level) hardcopy models and theories used in predicting the color of color halftones include the Clapper–Yule model equation,[15] the Beer–Bouguer law,[16] and the Kubelka–Munk theory.[17]

In addition to the macroscopic models listed above, models that use the halftone microstructure information and model the color (tone) reproduction microscopically (at the dot level or even finer) have also been developed. One of the earliest and the most important works following a microscopic approach was carried out by Ruckdeschel and Hauser.[18] They defined the reflectance at given location (x, y) on the paper substrate, $R(x, y)$ as

$$R(x,y) = R_p(x,y)T(x,y)\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} T(x',y')G(x-x',y-y');\, dx'dy' \qquad (7.3)$$

where $R_p(x, y)$ = reflectance of the paper substrate
$T(x, y)$ = transmittance of the colorant layer
$G(x, y)$ = point spread function of the paper substrate

In this approach, the incident light is assumed to pass through the colorant with no scattering, be partially absorbed, then scatter in the paper substrate and exit by passing through the colorant layer, where it is partially absorbed once more. Other similar approaches include the dot gain modeling by Gustavson,[19] the probabilistic framework developed by Arney and Katsube,[20,21] the probabilistic scattering model for clustered-dot halftones proposed by Huntsman,[22] the model based on photon transport and diffusion by Rogers,[23] the model incorporating light scattering and ink spreading devised by Emmel and Hersch,[24] the three-dimensional model developed by Yang and Kruse,[25] and the generalization of the Ruckdeschel and Hauser model by Agar.[26] Further details about these models can be found in Chapter 3.

The second group of color hardcopy model-based approaches used in color halftoning employs a dot model that characterizes nonideal dot profiles and interactions between neighboring dots. One of the most widely used dot models for monochrome printers is the hard circular dot model introduced by Roetling and Holladay.[27] This popular model is an idealization of printer behavior, and it assumes that every printer dot is a perfect circle as depicted in Figure 7.3. The circular dot modeling approach is easily generalizable to color halftones. Pappas[28] generalized the circular dot model to three-colorant printers where the color of a given pixel in the halftone is expressed as a sum of the overlapped colors of the segments, weighted by the areas of the segments, written in terms of the parameters α, β, and γ as shown in Figure 7.3. He used this model for color error diffusion and an iterative least-squares-model-based color halftoning technique. This parametric circular color dot overlap model was also used by Kim et al.[29] They incorporated the model into color error diffusion in the device independent Commission Internationale de l'Eclairage (International Commission on

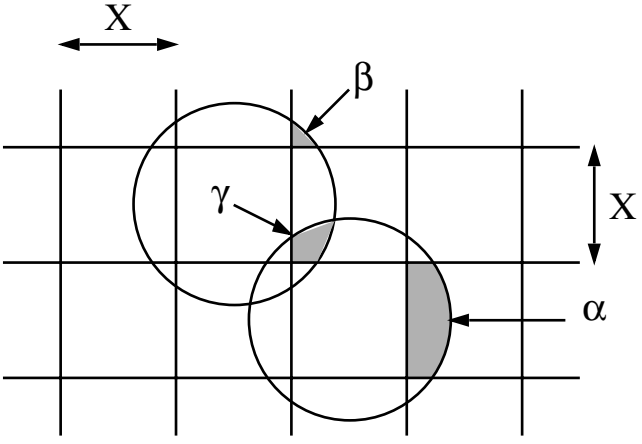


Figure 7.3 Hard circular dot model.

Illumination, CIE) *L*a*b** space and employed a Neugebauer color mixing model to find the average color of a target region in a printed test pattern made up of segments displaying different overlapped colors. Lee et al.[30] used a similar area coverage LUT-based color dot interaction model to modify the dot diffusion halftoning algorithm introduced by Knuth.[31] Lai and Chen[32] exploited the circular dot model presented[28] for vector color error diffusion in YCrCb space, and they also incorporated edge enhancement. Crounse[33] presented the generalization and analyzed the limitations of the same dot model.

Another novel dot-interaction model first introduced for monochrome printers and later extended to color printers is the 2×2 centering model, which was developed by Wang et al.[34] and generalized to color by Wang.[35] In this model, the printer grid is assumed to be offset from the pixel grid for the continuous-tone image to be halftoned. This results in a structure wherein four (2×2) neighboring printer dots in the printer grid contribute to the rendered tone (color) of one image pixel as shown in Figure 7.4. The parameters of the model for the monochrome (color) case are the reflectances of all possible 2×2 monochrome (color) overlap patterns. The Yule–Nielsen effect can also be incorporated into this model.[36] We will discuss the use of this model in detail in Section 7.4.3.

The third group of color hardcopy characterization approaches used in color halftoning is motivated by the frequently used technique in monochrome halftoning called *tone correction*.[27,37] This approach precompensates for dot interactions by modifying the image data prior to halftoning and printing, using a LUT of input absorptance vs. printed absorptance. In color halftoning, the method theoretically can be applied by halftoning and printing all possible colors, making colorimetric measurements of the resulting halftones, and applying the inverse mapping of this transformation to color images prior to halftoning to obtain halftones with desired colorimetric values. Because measuring halftone patches for a large set of colors (e.g., $256^3 \approx 16{,}750{,}000$ for an 8-bit color imaging system) is not feasible, it is impossible to have a complete *color correction* LUT. Instead, a limited number of halftoned colors are stored in a LUT, and interpolation schemes are employed to obtain the colorimetric values for the remaining colors. See Chapter 5 for additional details.

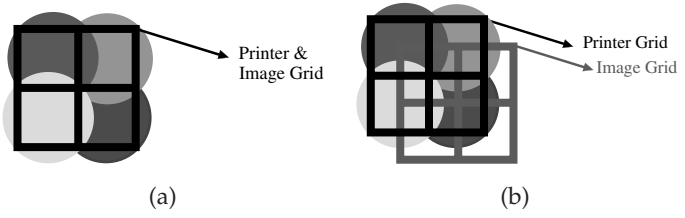


Figure 7.4 ***(See color insert)*** Alignment of the printer and image grids: (a) conventional method and (b) 2×2 centering method.

7.3 Color human visual system models

Psychophysical studies prove that color/tone discrimination and appearance closely depend on spatial pattern in addition to the global differences in perceptual attributes such as hue, saturation, and brightness.[38,39] This finding has motivated the use of color human visual system (HVS) models for design and evaluation of the performance of digital monochrome and color halftoning algorithms that aim to minimize the visual differences between continuous-tone images and their halftones.

In monochrome halftoning, the HVS models that have been incorporated into halftoning algorithms have been limited to linear shift-invariant filters based on the contrast sensitivity function (CSF) of the human viewer.[40] In color halftoning, different modeling approaches have been used. Pappas[28] employed a two-dimensional filter to model the HVS characteristics, which he called the "eye filter." This two-dimensional finite impulse response (FIR) filter was obtained as a separable combination of one-dimensional approximations to the spatial frequency sensitivity function of the HVS, estimated by Manos and Sakrison.[41] He applied the same filter to all the color components (red, green, and blue) in his halftoning algorithm and minimized the least-squares error metric independently for each color component. Other researchers have employed models based on representing the output of the human visual system with one achromatic channel and two opponent-color chromatic channels, a feature common to many of the color appearance models and color vision models.[42,43] Mulligan was perhaps the first to propose that color halftone textures could be made less visible by selectively partitioning the error between the continuous-tone original and its halftone reproduction into achromatic and chromatic channels.[44,45]

Zhang et al. used an opponent-channel-representation-based HVS model called S-CIELAB, proposed by Zhang and Wandell,[43] to predict the perceived quality of color halftone images.[46] In S-CIELAB, a spatial extension of CIELAB, a color image is transformed into an opponent-color space and decomposed into a luminance, a red–green (*R-G*), and a blue–yellow (*B-Y*) image. Each opponent-color image is filtered by a different two-dimensional separable spatial kernel of the form

$$f(x,y) = k\sum_i w_i E_i(x,y) \tag{7.4}$$

where

$$E_i(x,y) = k_i \exp\left(\frac{-(x^2+y^2)}{\sigma_i}\right) \tag{7.5}$$

and x, y, and σ_i are in degrees of visual angle. In the discrete implementation, the scale factor k_i is chosen so that E_i sums to 1. The scale factor k is chosen

so that, for each color plane, its two-dimensional kernel f sums to one. The parameters (w_i, σ_i) for the three opponent-color images are summarized in Table 7.1. After filtering each opponent-color image, the filtered representation is transformed to a CIE XYZ representation, and the S-CIELAB values are computed using the standard CIE $L^*a^*b^*$ formulae.

Table 7.1 Parameters for the S-CIELAB HVS Model

	w_i	σ_i
Luminance	0.921	0.0283
Luminance	0.105	0.133
Luminance	–0.108	4.336
Red–green	0.531	0.0392
Red–green	0.330	0.494
Blue–yellow	0.488	0.0536
Blue–yellow	0.371	0.386

Agar and Allebach[47] embedded a luminance/chrominance-based HVS model developed by Flohr et al.[48] into an iterative halftoning scheme. Lin and Allebach[49] exploited the same model within a least-squares search algorithm to design color screens. This model, which we will refer to as the Flohr HVS model,[48] is based on the uniform color space CIE $L^*a^*b^*$. The nonlinear transformation from CIE XYZ tristimulus values to CIE $L^*a^*b^*$ values cascaded with the linear transformation from red, green, blue (RGB) image coordinates to CIE XYZ tristimulus values does not preserve the spatially averaged tones of images,[50] which is undesirable in halftone color reproduction. Therefore, Flohr et al.[48] linearized the CIE $L^*a^*b^*$ color space about the reference stimulus (X_n, Y_n, Z_n). They denoted this linearized color space by $Y_yC_xC_z$, and defined it as follows:

$$
\begin{aligned}
Y_y &= 116\frac{Y}{Y_n} \\
C_x &= 200\left[\frac{X}{X_n} - \frac{Y}{Y_n}\right] \\
C_z &= 500\left[\frac{Y}{Y_n} - \frac{Z}{Z_n}\right]
\end{aligned}
\tag{7.6}
$$

The Y_y component serves as a correlate of luminance, and the C_x and C_z components are similar to the red–green (R-G) and blue–yellow (B-Y) opponent-color chrominance components that Mullen studied.[51]

In the linear, channel-independent, and shift-invariant Flohr HVS model,[48] the luminance channel combines models presented by Nasanen[52]

and Sullivan et al.[53] The luminance spatial frequency response $H_{Y_y}(\bar{u}, \bar{v})$ is given by

$$H_{Y_y}(\bar{u},\bar{v}) = aL^b \exp\left(\frac{(180)\sqrt{\bar{u}^2 + \bar{v}^2}}{\pi[c\ln(L) + d]}\right) \tag{7.7}$$

where $a = 131.6$
$b = 0.3188$
$c = 0.525$
$d = 3.91$
L = average luminance of the light reflected from the print (in $\frac{cd/m^2}{2}$)
$(\bar{u},\bar{v})$ = spatial frequency coordinates in cycles/radian subtended at the retina

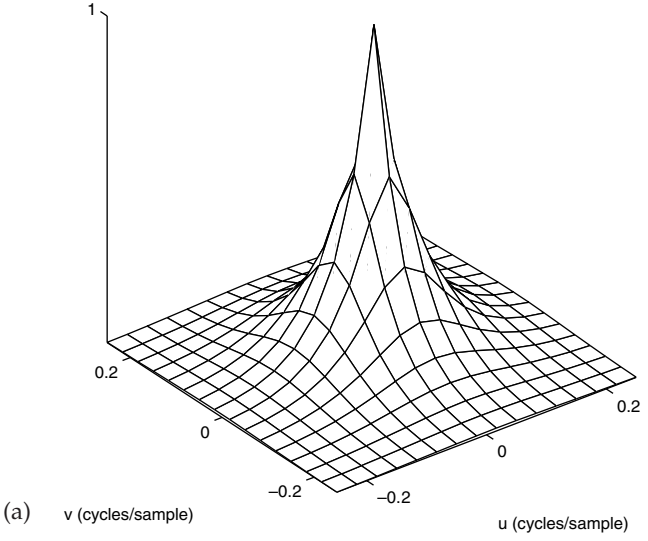
The chrominance spatial frequency response of the Flohr HVS model is based on an approximation by Kolpatzik and Bouman[54] to experimental data collected by Mullen.[51] This chrominance frequency response denoted by $H_{C_xC_z}(\bar{u},\bar{v})$ is common to both of the opponent-color chrominance components and is given by

$$H_{C_xC_z}(\bar{u},\bar{v}) = A\exp(-\alpha\sqrt{\bar{u}^2 + \bar{v}^2}) \tag{7.8}$$

where $\alpha = 0.419$
$A = 100$
$(\bar{u},\bar{v})$ = spatial frequency coordinates in cycles/radian subtended at the retina

Figure 7.5 shows the luminance and chrominance spatial frequency responses $H_{Y_y}(u, v)$, $H_{C_xC_z}(u, v)$ for 600-dpi prints viewed at 9 inches, where (u, v) are the spatial frequency coordinates in cycles/sample.

To enable adjustment of the relative importance between the luminance and chrominance errors in the error metric, the luminance frequency response is multiplied by a weighting factor, κ. As κ increases, more of the error is placed into the chrominance components. Flohr et al. employed this model in color image quantization[48] and showed that, as κ exceeds unity, the textures in the resulting halftones shift from being contrasty and monochromatic to being smoother and multicolored. They found that $\kappa = 4$ gives visually pleasing results. The same weighting factor was also used by Balasubramanian et al.[55] to form a visually weighted error metric in quantization of color images in a luminance/chrominance-based color space and by Kolpatzik and Bouman[54] to form a visually weighted error metric in error diffusion for color displays.



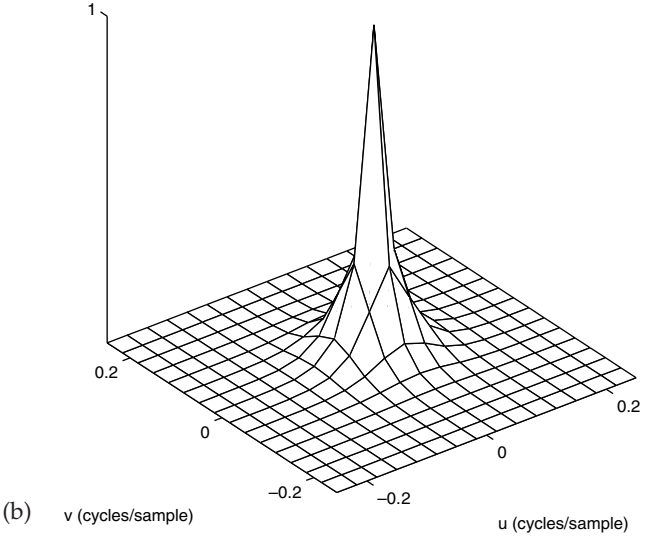


Figure 7.5 Spatial frequency responses for 600-dpi prints viewed at 9 inches: (a) luminance $\left|H_{Y_y} u, v\right|^2$ and (b) chrominance $\left|H_{C_x, C_z} u, v\right|^2$.

Taking the two-dimensional inverse continuous space Fourier transforms of $H_{Y_y}(\bar{u},\bar{v})$ and $H_{C_xC_z}(\bar{u},\bar{v})$ gives the impulse response functions $h_{Y_y}(\bar{x},\bar{y})$ and $h_{C_xC_z}(\bar{x},\bar{y})$ where the spatial coordinates $(\bar{x}, \bar{y})$ have units of radians subtended at the retina. For a printed page viewed from a distance D in inches, a length x in inches on the page corresponds to $tan^{-1}(x/D)$ radians subtended at the retina, and $tan^{-1}(x/D) \approx x/D$ radians, for $x \ll D$. Substituting this approximation into $h_{Y_y}(\bar{x}, \bar{y})$ and $h_{C_xC_z}(\bar{x}, \bar{y})$, we obtain the HVS

point spread functions $\tilde{p}_x(\mathbf{x})$ for $i = Y_y, C_x, C_z$, where $\mathbf{x} \equiv (x, y)$ now has units of inches measured on the print.

$$\tilde{p}_{Y_y}(x) = \frac{1}{D^2} h_{Y_y}\left(\frac{\mathbf{x}}{D}\right) = F^{-1}\{H_{Y_y}(D\bar{u}, D\bar{v})\} \tag{7.9}$$

$$\tilde{p}_{C_x}(\mathbf{x}) = \tilde{p}_{C_z}(\mathbf{x}) = \frac{1}{D^2} h_{C_x C_z}\left(\frac{\mathbf{x}}{D}\right) = F^{-1}\{H_{C_x C_z}(D\bar{u}, D\bar{v})\} \tag{7.10}$$

As the viewing distance D increases, the supports of the HVS point spread functions increase; i.e., a larger surrounding area affects our perception of a given point on the printed page.

The HVS models have been used in all three classes of digital color halftoning algorithms: the point-wise approaches such as screening, the neighborhood based approaches such as error diffusion, and the iterative approaches. In the next section, we discuss the HVS model-based iterative color halftoning algorithm. We present the application of HVS models to neighborhood-based approaches and to screen design in Section 7.5 and 7.6, respectively.

7.4 HVS model-based iterative color halftoning algorithms

In iterative color halftoning algorithms, several passes over the image pixels are made to minimize an error metric or satisfy certain constraints before the halftoning process is completed. In these algorithms, HVS models are commonly used to predict the perceived difference between the continuous-tone original image and its halftone in the error metric. A wide range of iterative approaches to digital halftoning were developed during the early 1990s.[56] In 1992, three groups simultaneously reported iterative approaches to minimizing frequency-weighted mean-squared error between the continuous-tone and halftone images.[57–59] This work forms the basis for the iterative halftoning algorithm to be developed in this section.

Pappas[28] developed a least-squares model-based color halftoning algorithm that employed both a color hardcopy model and a HVS model. He used the circular dot overlap model described in Section 7.2 and the two-dimensional eye filter described in Section 7.3. In this algorithm, starting with a totally white (or black) halftone or a halftone obtained using error diffusion, the pixels in the halftone are visited in some fixed (such as raster) or random order. For each pixel in the halftone, the optimal halftone state (colorant combination) that minimizes the squared error between the perceived value of the halftone pixel found using the circular dot overlap model and the two-dimensional eye filter, and the perceived value of the corresponding pixel in the original continuous-tone image found using the two-dimensional eye filter is selected. The iterations over the image pixels are

stopped when there are no more halftone state changes that reduce the perceived error. This algorithm results in a local minimum of the error metric, and the final local minimum depends on the initial choice of halftone used.

Another example of HVS model-based iterative color halftoning algorithms is the model-based color halftoning using direct binary search developed by Flohr et al.[48] and Agar and Allebach,[47] which uses the Flohr HVS model[48] described in Section 7.3. Below, we describe this dispersed dot color halftoning method in detail.

7.4.1 Preliminaries

We use $[\mathbf{m}] = [m, n]^T$ and $(\mathbf{x}) = (x, y)^T$ to denote discrete and continuous spatial coordinates in an image, respectively. We denote a continuous-tone color image by a three-dimensional vector valued function $\mathbf{f}_{\text{type}}[\mathbf{m}]$: $\mathcal{R}^2 \rightarrow \mathcal{R}^3$ where the input values are the discrete spatial coordinates in the image and the output values can be the *RGB*, *cmy*, *XYZ*, or $Y_y C_x C_z$ values at a given location in the image, depending on the subscript *type*. This function can be decomposed into three scalar valued functions f_i: $\mathcal{R}^2 \rightarrow \mathcal{R}$; i = *R*, *G*, *B*; *i* = *c*, *m*, *y*; *i* = *X*, *Y*, *Z*; or *i* = Y_y, C_x, C_z. Similarly, we denote a color halftone bitmap by a vector valued *L*-D function $\mathbf{g}_{type}[\mathbf{m}]$: $R^2 \rightarrow R^L$ where the input values are again the discrete spatial coordinates in the image, and *L* is the number of colorants that the color halftone contains. This function can again be decomposed into *L* scalar valued functions.

We restrict our analysis to bilevel three-colorant *CMY* printers and bilevel four-colorant *CMYK* printers where full undercolor removal is employed; i.e., at a given pixel (printer addressable point), a black dot replaces cyan, magenta, and yellow dots if and only if all three are present there. Therefore, we assume that $L = 3$. For the initial analysis, we also assume that the dot interactions between neighboring pixels in the printer grid are additive. As discussed in Section 7.2, this assumption is generally not true. The interaction of the colorants of the neighboring printer pixels with each other and the paper substrate is fairly complex. We will discuss how we incorporate a dot-interaction model into the color direct binary search halftoning algorithm in Section 7.4.3.

7.4.2 Color direct binary search

The color direct binary search (CDBS) halftoning method is an iterative search-based algorithm that minimizes a measure of perceived error by starting from an initial halftone and modifying the halftone locally. Figure 7.6 illustrates the overall structure of the algorithm. The initial halftone is obtained by halftoning the *R*, *G*, and *B* components of the continuous-tone color image separately using the same screen, which is obtained with the monochrome DBS halftoning method,[60,61] and then superimposing these three halftones under the assumption that $C = 1 - R$, $M = 1 - G$, $Y = 1 - B$. For the initial halftone, we compute the perceived error in the device-independent opponent-color space of $Y_y C_x C_z$ as suggested by Flohr et al.[48] and

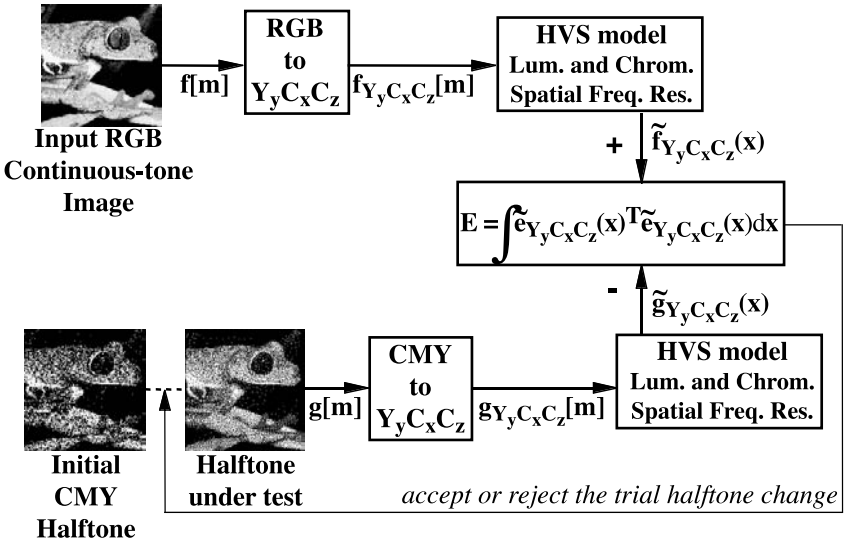


Figure 7.6 Block diagram for the color direct binary search (CDBS) algorithm.

discussed below. Then, we process the pixels of the halftone in a raster-scan order. At each pixel, we compute the effect on the error metric of toggling (changing the colorant combination) of the pixel and swapping it with its eight nearest neighbor (NN) pixels. We accept the trial change, if any, that most reduces the error metric, and we iterate until there are no more accepted trial changes, i.e., until the error metric reaches a local minimum.

To compute the perceived error, we first need to convert the original continuous-tone color image **f[m]** and the color halftone bitmap **g[m]** from the device-dependent color spaces of *RGB* and *CMY*, respectively, to the device-independent opponent-color space of $Y_yC_xC_z$ using transformations calibrated for a given color imaging system. To transform **g[m]** into $Y_yC_xC_z$, we use an eight-entry $CMY \rightarrow Y_yC_xC_z$ LUT. This LUT contains the $Y_yC_xC_z$ values of the eight possible colorant combinations known as the Neugebauer primaries (*C*, *M*, *Y*, *R*, *G*, *B*, *K*, *W*), measured with a spectrophotometer, for a given printer and paper substrate.

To transform **f[m]** into $Y_yC_xC_z$, we use a simple $RGB \rightarrow Y_yC_xC_z$ transformation employing a common undercolor removal (gray component replacement) strategy. This strategy is based on the assumption that the printing geometry is dot-on-dot; i.e., the colorants are placed exactly on top of each other. In this method, the colorants *C*, *M*, and *Y* are referred to as *subtractive primaries*. The colors corresponding to two-color overprints *R*(*M* + *Y*), *G*(*C* + *Y*), and *B*(*C* + *M*) are referred to as subtractive secondaries. The three-color overprint is called *K*, and the unprinted substrate is defined as *W*. The method computes the proportions of the eight Neugebauer primaries, *fi*, for *i* = *W*, *C*, *M*, *Y*, *R*, *G*, *B*, *K* for a given desired color in an image defined by a three-tuple (R_i, G_i, B_i) according to the following equations:

```
fW = min(R_i, G_i, B_i)
fK = 1 – max(R_i, G_i, B_i)
if R_i = min(R_i, G_i, B_i)
   fR = 0; fM = 0; fY = 0; fC = min(R_i,G_i) – fW; fG = G_i – fC – fW; fB =
   B_i – fC – fW
end
if G = min(R,G,B)
   fG = 0; fC = 0; fY = 0; fM = min(R_i,B_i) – fW; fR = R_i – fM – fW; fB =
   B_i – fM – fW
end
if B = min(R,G,B)
   fB = 0; fC = 0; fM = 0; fY = min(R_i,G_i) – fW; fR = R_i – fY – fW; fG =
   G_i – fY – fW
end
```

Because R_i, G_i, and B_i are color coordinates in an additive color system such as a display, *fW* is set equal to the minimum of R_i, G_i, B_i; *fK* is set equal to one minus their maximum. For example, if $B_i = min(R_i, G_i, B_i)$ as shown in Figure 7.7, then *fW* is set equal to B_i, and the proportion of the corresponding subtractive secondary *fB* is set equal to zero. Because we assume dot-on-dot printing geometry, this also implies that *fC* and *fM* are equal to zero. In this method, the desired color can always be created using only one subtractive secondary (*R*, for Figure 7.7), one subtractive primary (*Y*, for Figure 7.7), black, and white. This method necessitates the measurement of the $Y_yC_xC_z$ values of only the eight Neugebauer primaries. After *fi*, for $i = W, C, M, Y, R, G, B, K$ have been computed, the $Y_yC_xC_z$ coordinates of the desired color can be found as a weighted sum of the $Y_yC_xC_z$ values of the eight Neugebauer primaries, using the *fi* as weights. Because this method assumes that the color printer has three colorants and that the color black is obtained by putting down *C*, *M*, and *Y* together, referred to as *composite black*, caution should be taken in generalizing this strategy to four-colorant *CMYK* printers that use a black colorant to create the color black, referred to as *process black.*

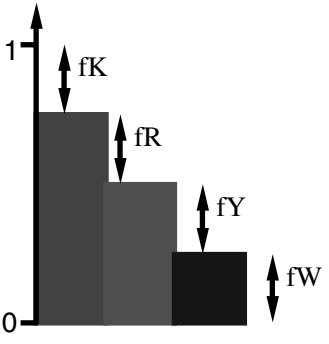


Figure 7.7 Undercolor removal strategy.

Generally, process black is noticeably darker than composite black and has a $Y_yC_xC_z$ value significantly closer to (0, 0, 0) than composite black. For *CMYK* printers, we have observed, as have others,[62,63] that favoring composite black dots over process black dots improves the color texture of the halftones. Therefore, for *CMYK* printers with full undercolor removal, we modify this undercolor removal strategy as follows: we decrease the proportion of the black color *fK* and linearly increase the proportions of the remaining Neugebauer primaries using Equation 7.11 so that the sum of *fi'* equals 1, as the physical area coverage of the Neugebauer primaries should sum to 1. We again find the $Y_yC_xC_z$ coordinates as a weighted sum of the $Y_yC_xC_z$ values of the eight Neugebauer primaries using the modified weights

$$fK' = fK^2, \quad fi' = \left(1 - \frac{fK'}{fK}\right) fi, i = W,C,\mathrm{M},Y,R,G,\mathrm{B} \tag{7.11}$$

We denote the $Y_yC_xC_z$ continuous-tone original color image and the $Y_yC_xC_z$ rendered color image by $\mathbf{f}_{Y_yC_xC_z}[\mathbf{m}]$ and $\mathbf{g}_{Y_yC_xC_z}[\mathbf{m}]$, and their components by $f_i[\mathbf{m}]$, $i = Y_y, C_x, C_z$ and $g_i[\mathbf{m}]$, $i = Y_y, C_x, C_z$, respectively. Note that we also represent the *R*, *G*, *B* components of a continuous-tone color image by $f_i[\mathbf{m}]$. The quantity that $f_i[\mathbf{m}]$ stands for will be clear from the context. We define the error image in the $Y_yC_xC_z$ color space and its components as

$$\begin{aligned} \mathbf{e}_{Y_yC_xC_z}[\mathbf{m}] &\equiv \mathbf{f}_{Y_yC_xC_z}[\mathbf{m}] - \mathbf{g}_{Y_yC_xC_z}[\mathbf{m}] \\ e_i[\mathbf{m}] &\equiv f_i[\mathbf{m}] - g_i[\mathbf{m}] \qquad i = Y_y, C_x, C_z \end{aligned} \tag{7.12}$$

Using the linear, channel-independent, and shift-invariant Flohr HVS model introduced in Section 7.3, and assuming additive interaction between neighboring dots, we model the perceived error $\tilde{\mathbf{e}}_{Y_yC_xC_z}(\mathbf{x})$ in the $Y_yC_xC_z$ opponent-color space, which is defined on the continuous spatial coordinates ($\mathbf{x}$) as

$$\begin{aligned} \tilde{\mathbf{e}}_{Y_yC_xC_z}(\mathbf{x}) = \sum_{\mathbf{m}} \mathrm{diag}(&\tilde{p}_{dot\_Y_y}(\mathbf{x} - \mathbf{Xm}), \\ &\tilde{p}_{dot\_C_x}(\mathbf{x} - \mathbf{Xm}), \tilde{p}_{dot\_C_y}(\mathbf{x} - \mathbf{Xm}))\mathbf{e}_{Y_yC_xC_z}[\mathbf{m}] \end{aligned} \tag{7.13}$$

where $\tilde{p}_{dot\_i}(\mathbf{x}) \equiv \tilde{p}_i(\mathbf{x}) * p_{dot}(\mathbf{x})$ is the HVS point spread function for the *i*th component of the $Y_yC_xC_z$ opponent-color space $\tilde{p}_i(\mathbf{x})$ convolved with the printer dot profile $p_{dot}(\mathbf{x})$, $\mathbf{X}$ is a periodicity matrix whose columns form the basis for the lattice of printer addressable dots, and diag(·) is a diagonal matrix with the diagonal elements listed between the parentheses. With a printer for which the lattice of addressable points is rectangular with horizontal and vertical spacing X, $\mathbf{X}$ = diag(X, X). Because the printer dot profile has much more limited support than the HVS point spread function, and we assume that the printer dot profile has unit volume, then $\tilde{p}_{dot\_i}(\mathbf{x}) \approx \tilde{p}_i(\mathbf{x})$. Therefore, we can rewrite Equation 7.13 as

$$\tilde{\mathbf{e}}_{Y_y C_x C_z}(\mathbf{x}) = \sum_{\mathbf{m}} \tilde{\mathbf{P}}(\mathbf{x} - \mathbf{X}\mathbf{m}) \mathbf{e}_{Y_y C_x C_z}[\mathbf{m}] \tag{7.14}$$

where

$$\tilde{\mathbf{P}}(\mathbf{x}) \equiv \text{diag}(\tilde{p}_{Y_y}(\mathbf{x}), \tilde{p}_{C_x}(\mathbf{x}), \tilde{p}_{C_z}(\mathbf{x})) \tag{7.15}$$

We define the error metric E to be the sum of the total squared perceived errors in all three components of the $Y_y C_x C_z$ color space.

$$E = \int \tilde{\mathbf{e}}_{Y_y C_x C_z}(\mathbf{x})^T \tilde{\mathbf{e}}_{Y_y C_x C_z}(\mathbf{x}) \text{dx} \tag{7.16}$$

Substituting Equation 7.14 into Equation 7.16, we get

$$E = \sum_{\mathbf{m}} \sum_{\mathbf{n}} \mathbf{e}_{Y_y C_x C_z}[\mathbf{m}]^T \left(\int \tilde{\mathbf{P}}(\mathbf{x} - \mathbf{X}\mathbf{m}) \tilde{\mathbf{P}}(\mathbf{x} - \mathbf{X}\mathbf{n}) \text{dx} \right) \mathbf{e}_{Y_y C_x C_z}(\mathbf{n}) \tag{7.17}$$

We denote the matrix of autocorrelation functions of $\tilde{p}_i(\mathbf{x})$, for $i = Y_y, C_x, C_z$, by $\mathbf{c}_{\tilde{p}\tilde{p}}(\mathbf{x})$, and the matrix of cross-correlation functions between $\tilde{p}_i(\mathbf{x})$ and $\tilde{e}_i(\mathbf{x})$, for $i = Y_y, C_x, C_z$ by $\mathbf{c}_{\tilde{p}\tilde{e}}(\mathbf{x})$.

$$\mathbf{c}_{\tilde{p}\tilde{p}}(\mathbf{x}) \equiv \text{diag}\left(\int \tilde{p}_{Y_y}(\mathbf{y}) \tilde{p}_{Y_y}(\mathbf{y} + \mathbf{x}) dy, \int \tilde{p}_{C_x}(\mathbf{y}) \tilde{p}_{C_x}(\mathbf{y} + \mathbf{x}) dy, \int \tilde{p}_{C_z}(\mathbf{y}) \tilde{p}_{C_z}(\mathbf{y} + \mathbf{x}) \mathbf{dy} \right) \tag{7.18}$$

$$\mathbf{c}_{\tilde{p}\tilde{e}}(\mathbf{x}) \equiv \text{diag}\left(\int \tilde{p}_{Y_y}(\mathbf{y}) \tilde{e}_{Y_y}(\mathbf{y} + \mathbf{x}) dy, \int \tilde{p}_{C_x}(\mathbf{y}) \tilde{e}_{C_x}(\mathbf{y} + \mathbf{x}) dy, \int \tilde{p}_{C_z}(\mathbf{y}) \tilde{e}_{C_z}(\mathbf{y} + \mathbf{x}) \mathbf{dy} \right) \tag{7.19}$$

Evaluating these two matrices at the pixels in the printer grid $\mathbf{x} = \mathbf{X}\mathbf{m}$ gives us $\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m}]$ and $\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}]$. Then, we can rewrite Equation 7.17 as

$$E = \sum_{\mathbf{m}} \sum_{\mathbf{n}} \mathbf{e}_{Y_y C_x C_z}[\mathbf{m}]^T \mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m} - \mathbf{n}] \mathbf{e}_{Y_y C_x C_z}[\mathbf{n}] = \sum_{\mathbf{m}} \mathbf{e}_{Y_y C_x C_z}[\mathbf{m}]^T \mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}] \tag{7.20}$$

We use an efficient technique for calculating the effect of trial halftone changes (toggles and swaps) on the error metric E, a generalization of a technique developed by Analoui and Allebach[60] and further refined by Lieberman and Allebach[61] for monochrome halftoning using DBS, which we describe below.

For a bilevel three-colorant CMY printer, each halftone pixel displays one of eight (2^3) possible colorant combinations. These combinations correspond to one-, two-, and three-color overprints of the colorants *C*, *M*, *Y* or to no colorant on paper (*W*). An alternative labeling of these combinations is therefore *W*, *C*, *M*, *Y*, *MY*, *CY*, *MY*, and *CMY*, respectively. Similarly, for a binary four-colorant printer with full undercolor removal, there are also eight

possible combinations, which are labeled *W*, *C*, *M*, *Y*, *MY*, *CY*, *MY*, and *K*, respectively. Therefore, a toggle for CDBS is defined as changing the value of a halftone pixel from a given colorant combination to a different one out of these eight.

Consider a trial halftone pixel toggle at index $\mathbf{m}_t$. We denote the change that this toggle will cause in the $Y_yC_xC_z$ color rendered image $\mathbf{g}_{Y_yC_xC_z}[\mathbf{m}]$ at index $\mathbf{m}_t$ by $\mathbf{a}[\mathbf{m}_t] = [a_{Y_y}[\mathbf{m}_t], a_{C_x}[\mathbf{m}_t], a_{C_z}[\mathbf{m}_t]]^T$. We then define $\mathbf{A}[\mathbf{m}_t] = \mathrm{diag}(\mathbf{a}[\mathbf{m}_t])$. With this toggle, $\mathbf{g}_{Y_yC_xC_z}[\mathbf{m}]$, $\mathbf{e}_{Y_yC_xC_z}[\mathbf{m}]$, and $\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}]$ will change as follows:

$$\mathbf{g'}_{Y_yC_xC_z}[\mathbf{m}] = \mathbf{g}_{Y_yC_xC_z}[\mathbf{m}] + \mathbf{a}[\mathbf{m}_t]\delta[\mathbf{m} - \mathbf{m}_t] \tag{7.21}$$

$$\mathbf{e'}_{Y_yC_xC_z}[\mathbf{m}] = \mathbf{e}_{Y_yC_xC_z}[\mathbf{m}] + \mathbf{a}[\mathbf{m}_t]\delta[\mathbf{m} - \mathbf{m}_t] \tag{7.22}$$

$$\mathbf{c'}_{\tilde{p}\tilde{e}}[\mathbf{m}] = \mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}] + \mathbf{A}[\mathbf{m}_t]\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m} - \mathbf{m}_t] \tag{7.23}$$

where the prime denotes the version after the toggle. The change ΔE_t in E due to the toggle can then be written as

$$\begin{aligned}
\Delta E_t &= \sum_{\mathbf{m}} \mathbf{e'}_{Y_yC_xC_z}{}^T[\mathbf{m}]\mathbf{c'}_{\tilde{p}\tilde{e}}[\mathbf{m}] - \sum_{\mathbf{m}} \mathbf{e}_{Y_yC_xC_z}{}^T[\mathbf{m}]\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}] \\
&= \sum_{\mathbf{m}} (\mathbf{e}_{Y_yC_xC_z}{}^T[\mathbf{m}] - \mathbf{a}[\mathbf{m}_t]\delta[\mathbf{m} - \mathbf{m}_t])^T(\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}] + \mathbf{A}[\mathbf{m}_t]\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m} - \mathbf{m}_t]) \\
&\quad - \sum_{\mathbf{m}} \mathbf{e}_{Y_yC_xC_z}{}^T[\mathbf{m}]\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}] \\
&= \mathbf{a}[\mathbf{m}_t]^T\mathbf{A}[\mathbf{m}_t]\mathbf{c}_{\tilde{p}\tilde{p}}[0] + 2\mathbf{a}[\mathbf{m}_t]^T\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}_t]
\end{aligned} \tag{7.24}$$

where we have used $\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}] = \Sigma_{\mathbf{n}}\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m} - \mathbf{n}]\mathbf{e}_{Y_yC_xC_z}[\mathbf{n}]$ and $\mathbf{c}_{\tilde{p}\tilde{p}}[-\mathbf{m}] = \mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m}]$. The computational cost of evaluating ΔE_t can be reduced significantly by storing $\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m}]$ and $\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}]$ in LUTs. If the toggle under trial is accepted, then we update only the $\mathbf{g}_{Y_yC_xC_z}[\mathbf{m}]$ and $\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}]$ LUTs using Equations 7.21 and 7.23, respectively.

Another type of trial halftone change we will analyze is swapping the colorant combination of a given pixel in the halftone with that of another pixel. A swap is a more subtle change to the halftone under test than a toggle. By definition, it conserves the average local "color" in the area containing the swapped pixels and can be interpreted as two consecutive toggles. Consider a trial swap of the halftone pixels at indices $\mathbf{m}_t$ and $\mathbf{m}_s$. Let us denote the changes that the swap under trial will cause in the $Y_yC_xC_z$ rendered color image at indices $\mathbf{m}_t$ and $\mathbf{m}_s$ by $\mathbf{a}[\mathbf{m}_t]$ and $\mathbf{a}[\mathbf{m}_s]$, respectively. Similar to $\mathbf{A}[\mathbf{m}_t]$,

we define $\mathbf{A}[\mathbf{m}_s] \equiv \text{diag}(\mathbf{a}[\mathbf{m}_s])$. Then, the change in the error metric ΔE_s due to the swap can be written as

$$\Delta E_s = \Delta E_t + \mathbf{a}[\mathbf{m}_s]^T \mathbf{A}[\mathbf{m}_s] \mathbf{c}_{\tilde{p}\tilde{p}}[0] + 2\mathbf{a}[\mathbf{m}_s]^T \mathbf{c}'_{\tilde{p}\tilde{e}}[\mathbf{m}_s] \tag{7.25}$$

where ΔE_t is the change in the error metric due to the toggle of the halftone pixel at $\mathbf{m}_t$ to the colorant combination at $\mathbf{m}_s$, and $\mathbf{c}'_{\tilde{p}\tilde{e}}[\mathbf{m}_s]$ is the cross-correlation vector after this toggle, evaluated at $\mathbf{m} = \mathbf{m}_s$. Substituting Equations 7.23 and 7.24 into Equation 7.25, we obtain

$$\begin{aligned} \Delta E_s = {} & (\mathbf{a}[\mathbf{m}_t]^T \mathbf{A}[\mathbf{m}_t] + \mathbf{a}[\mathbf{m}_s]^T \mathbf{A}[\mathbf{m}_s]) \mathbf{c}_{\tilde{p}\tilde{p}}[0] + 2\mathbf{a}[\mathbf{m}_t]^T \mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}_t] \\ & + 2\mathbf{a}[\mathbf{m}_s]^T \mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}_s] + 2\mathbf{a}[\mathbf{m}_s]^T \mathbf{A}[\mathbf{m}_t] \mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m}_s - \mathbf{m}_t] \end{aligned} \tag{7.26}$$

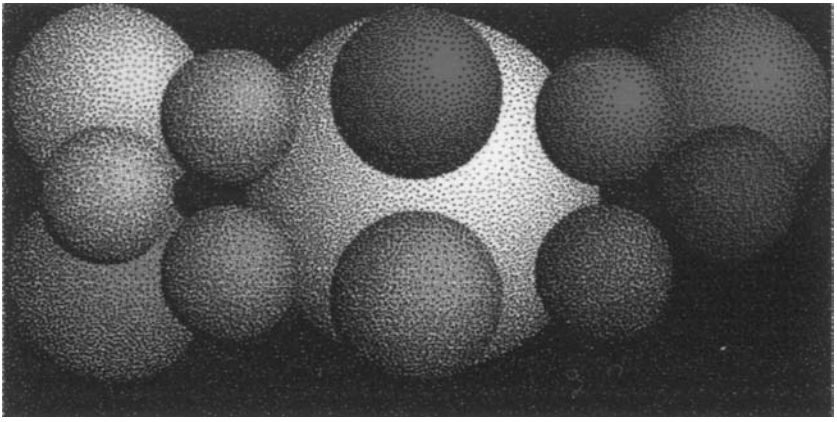
Similar to ΔE_t, the computational cost of finding ΔE_s can also be reduced greatly by keeping LUTs for $\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m}]$ and $\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}]$. If the swap under trial is accepted, the $\mathbf{g}_{Y_y C_x C_z}[\mathbf{m}]$ and $\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}]$ LUTs are updated as follows:

$$\mathbf{g}'_{Y_y C_x C_z}[\mathbf{m}] = \mathbf{g}_{Y_y C_x C_z}[\mathbf{m}] + \mathbf{a}[\mathbf{m}_t]\delta[\mathbf{m} - \mathbf{m}_t] + \mathbf{a}[\mathbf{m}_s]\delta[\mathbf{m} - \mathbf{m}_s] \tag{7.27}$$

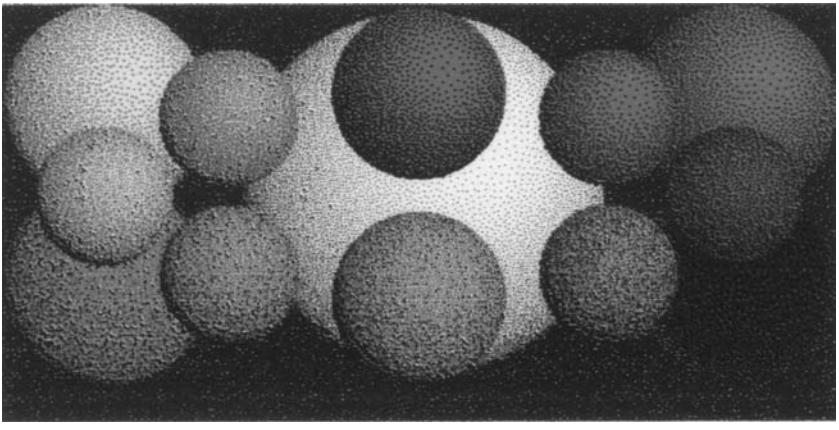
$$\mathbf{c}'_{\tilde{p}\tilde{e}}[\mathbf{m}] = \mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}] + \mathbf{A}[\mathbf{m}_t]\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m} - \mathbf{m}_t] + \mathbf{A}[\mathbf{m}_s]\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m} - \mathbf{m}_s] \tag{7.28}$$

In a recent paper, Lieberman and Allebach[61] showed that the time complexity of monochrome DBS can be improved considerably by using two efficient search techniques instead of computing the effect of every possible toggle and every possible NN swap in raster-scan order. We generalize and apply these two strategies to CDBS. First, we limit the search of possible swapping pixels to those in the anticausal neighborhood of a given pixel only and hence decrease the number of trial swap computations by 50%. Second, we exploit the fact that the cost of a trial change computation is much lower than that of the actual update if the trial change is accepted. Pixels of the halftone image are partitioned into $C \times C$ cells. These cells are processed in raster-scan order. Within each cell, the effect of all possible toggles and swaps are computed, but only one halftone pixel change is allowed for the whole cell. Only the one toggle or swap within the cell that causes the largest decrease in the error metric is accepted. We found empirically that $C = 3$ gives the lowest time complexity for CDBS.

Figure 7.8 compares two halftones of the *faxballs* image, one obtained by applying CDBS in the *RGB* space and the other by applying CDBS in the $Y_y C_x C_z$ space. To apply the CDBS halftoning algorithm in the *RGB* color space instead of $Y_y C_x C_z$, we use $\tilde{p}_{Y_y}(\mathbf{x})$, the HVS point spread function for the luminance channel, for all three color channels; we set $\kappa = 1$; and we choose our error metric to be the sum of total squared *RGB* errors after passing through this HVS filter. This approach is similar to the one presented by



(a)



(b)

Figure 7.8 ***(See color insert)*** 600-dpi scans of the 300-dpi halftones of the *faxballs* image obtained using CDBS (a) in the *RGB* space and (b) in the $Y_yC_xC_z$ space, with the simple $RGB \rightarrow Y_yC_xC_z$ calibration.

Pappas[28] for nonseparable printer models. The two 600-dpi scans of the 300-dpi halftones, printed on a Hewlett Packard 692C printer, display the difference in halftone texture. This is a CMYK inkjet printer with full undercolor removal so, at any pixel location where the halftoning algorithm puts dots of all three colorants cyan, magenta, and yellow on top of each other, these three dots are replaced by a single black dot. The comparison of these two scans shows that the use of a luminance/chrominance-based space such as $Y_yC_xC_z$ gives us an overall finer (and therefore more pleasing) texture. The differences in the visual consequences of the chosen color space are especially noticeable in highlights, such as the white ball, and all light shadow regions, such as the perimeter of the yellow ball and the light green ball that overlaps it.

Even though texture rendition is improved by the choice of a luminance/chrominance-based space instead of a device-dependent color space, there is room for further improvement, especially in the darker shadow areas of Figure 7.8b, where there are artifacts due to the interactions between dark dots neighboring light dots, e.g., in the shadow regions of the green ball and the cyan ball overlapping the lower part of the white ball. These artifacts are a result of the lack of a printer model in the CDBS halftoning algorithm and the invalidity of the assumptions made in the analysis above about the printer dots, i.e., that the dot profiles are identical for all colorants and that dot interactions are additive. In the next subsection, we will explain how we incorporate the measurement-based 2×2 centering printer dot interaction model[34,35] into CDBS to prevent the artifacts due to dot overlap and to improve color texture quality.

7.4.3 *Two-by-two centering-based CDBS*

The 2×2 centering printer dot interaction model[34,35] discussed in Section 7.2 provides a means of quantifying dot overlap in color halftones with only 1072 measurements. The novelty of this method is the idea that the printer grid is assumed to be shifted by (0.5, 0.5) pixels from the image grid as shown in Figure 7.4b. When the image grid is imposed on top of the printer grid as depicted in Figure 7.4b, every pixel in the image grid has the same rendered "color" that is created by the four (2×2) printer pixels and the overlaps in between these four printer pixels. Because there are eight possible colorant combinations per printer pixel for the three-colorant bilevel printers or four-colorant bilevel printers with full undercolor removal that we are modeling, this approach results in 8^4 (4096) possible (2×2) colorant combinations (i.e., "colors") for each rendered image pixel instead of the eight that the conventional method provides (Figure 7.4a). The assumption that the dots are symmetric with respect to their vertical and horizontal axes decreases the number of unique 2×2 colorant combinations to 1072. Calibration of the model necessitates the colorimetric measurement of these 1072 overlapping patterns. Symmetry characteristics of the pattern allows us to tile the 2×2 colorant combinations vertically and horizontally to form a color patch with an average XYZ tristimulus value equal to the XYZ tristimulus value of this combination as shown in Figure 7.9. We macro-

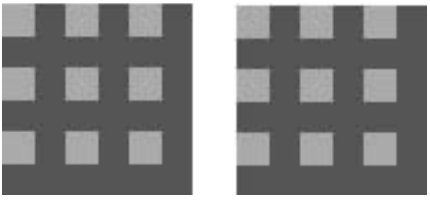


Figure 7.9 ***(See color insert)*** Two examples of the 2×2 colorant combination bitmaps.

scopically measure the XYZ tristimulus value directly using a spectrophotometer.

Macroscopic measurement of the colorant combinations allows us to quantify the dot overlap in these patterns and to measure the overlapped rendered color of the four colorant combinations making up the pattern. These 1072 measurements depend on both the colorant set and the paper substrate. We convert these measured XYZ values to $Y_yC_xC_z$ and store them in a LUT to incorporate into our CDBS halftoning algorithm, as was done by Pappas et al.[64] with a LUT of reflectances of all unique 3×3 pixel neighborhoods for monochrome error diffusion.

Incorporating the 2×2 centering method[34,35] into CDBS requires changes in the definition of the $Y_yC_xC_z$-rendered color image and the computation of the effect of trial halftone changes on the error metric. In 2×2 centering-based CDBS, the $Y_yC_xC_z$ centering based rendered color image at pixel location $\mathbf{m}$, which we will denote by $\mathbf{g}^o_{Y_yC_zC_z}[\mathbf{m}]$, is no longer a function of only $\mathbf{g}[\mathbf{m}]$; it is a function of the colorant combinations at the four printer pixels contributing to the rendered color at this pixel (Figure 7.4b). We denote the neighborhood of these four printer pixels that intersect with pixel $\mathbf{m}$ in the image pixel by $\mathcal{N}(\mathbf{m})$. Therefore, in centering-based CDBS

$$\mathbf{g}^o{}_{Y_yC_xC_z}[\mathbf{m}] \equiv O_{Y_yC_xC_z}(g[\mathbf{m}], \mathbf{m} \in \mathcal{N}(\mathbf{m})) \tag{7.29}$$

where $O_{Y_yC_xC_z}$ is the function that generates the rendered $Y_yC_xC_z$ for a given input 2×2 overlapping colorant combination using the LUT of the measured $Y_yC_xC_z$ values for the 4096 such combinations. We define a dual neighborhood $\mathcal{N}^{-1}(\mathbf{m})$ in the image grid which denotes the neighborhood made of four image pixels that intersect with pixel $\mathbf{m}$ in the printer grid, i.e., the four image pixels that are affected by the choice of the colorant combination in the halftone at pixel $\mathbf{m}$ in the printer grid.

For the 2×2 centering-based CDBS, the error image in the $Y_yC_xC_z$ color space $\mathbf{e}_{Y_yC_zC_z}[\mathbf{m}]$ is given by

$$\mathbf{e}_{Y_yC_zC_z}[\mathbf{m}] \equiv \mathbf{f}_{Y_yC_zC_z}[\mathbf{m}] - \mathbf{g}^o{}_{Y_yC_xC_z}[\mathbf{m}] \tag{7.30}$$

The remainder of the equations defining the error metric for CDBS (Equations 7.13 through 7.20) also apply to the 2×2 centering-based CDBS. The modifications to the computation of the effect on the error metric of the trial toggles and swaps are described below.

In centering-based CDBS, a trial toggle at halftone pixel $\mathbf{m}_t$ from a given colorant combination to a different one out of the eight possible colorant combinations causes a change in the $Y_yC_xC_z$-rendered color image at the four image pixel locations in $\mathcal{N}^{-1}(\mathbf{m}_t)$ as shown in Figure 7.10a. This trial toggle will change the $Y_yC_xC_z$-rendered color image by $\mathbf{a}[\mathbf{n}]$ at the pixels $\mathbf{n} \in \mathcal{N}^{-1}(\mathbf{m}_t)$. Similar to $\mathbf{A}[\mathbf{m}_t]$, we define $\mathbf{A}[\mathbf{n}] \equiv \text{diag}(\mathbf{a}[\mathbf{n}])$. This trial toggle will affect $\mathbf{g}^o{}_{Y_yC_xC_z}[\mathbf{m}]$, $\mathbf{e}_{Y_yC_xC_z}[\mathbf{m}]$, and $\mathbf{c}_{\tilde{p}\tilde{c}}[\mathbf{m}]$ as follows:

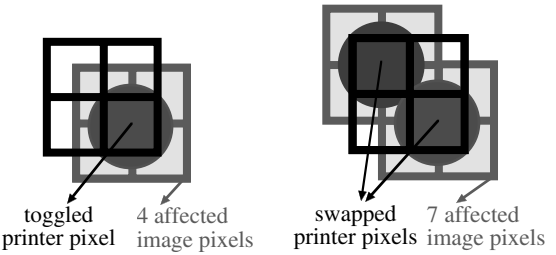


Figure 7.10 ***(See color insert)*** Effect of a (a) trial toggle and (b) trial swap.

$$\mathbf{g}^{o'}_{Y_yC_xC_z}[\mathbf{m}] = \mathbf{g}^{o}{}_{Y_yC_xC_z}[\mathbf{m}] + \sum_{\mathbf{n} \in \mathcal{N}^{-1}(\mathbf{m}_t)} \mathbf{a}[\mathbf{n}]\delta[\mathbf{m}-\mathbf{n}] \tag{7.31}$$

$$\mathbf{e}'_{Y_yC_xC_z}[\mathbf{m}] = \mathbf{e}_{Y_yC_xC_z}[\mathbf{m}] + \sum_{\mathbf{n} \in \mathcal{N}^{-1}(\mathbf{m}_t)} \mathbf{a}[\mathbf{n}]\delta[\mathbf{m}-\mathbf{n}] \tag{7.32}$$

$$\mathbf{c}'_{\tilde{p}\tilde{e}}[\mathbf{m}] = \mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}] + \sum_{\mathbf{n} \in \mathcal{N}^{-1}(\mathbf{m}_t)} \mathbf{A}[\mathbf{n}]\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m}-\mathbf{n}] \tag{7.33}$$

where the prime denotes the version after the toggle. We observe that, for a given trial toggle, centering-based CDBS changes a larger area in the $Y_yC_xC_z$-rendered color image than regular CDBS does, affecting four image pixels instead of one. However, the changes $\Delta Y_yC_xC_z$ observed at these four locations $\{\mathbf{a}[\mathbf{n}]|\mathbf{n} \in \mathcal{N}^{-1}(\mathbf{m}_t)\}$ are generally smaller in magnitude than we see at the single pixel location for regular CDBS. The change in E due to this trial toggle can be written as

$$\begin{aligned}
\Delta E_t &= \sum_{\mathbf{m}} \mathbf{e}'_{Y_yC_xC_z}{}^{T}[\mathbf{m}]\mathbf{c}'_{\tilde{p}\tilde{e}}[\mathbf{m}] - \sum_{\mathbf{m}} \mathbf{e}_{Y_yC_xC_z}{}^{T}[\mathbf{m}]\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}] \\
&= \sum_{\mathbf{m}} \left(\mathbf{e}_{Y_yC_xC_z}[\mathbf{m}] + \sum_{\mathbf{n} \in \mathcal{N}^{-1}(\mathbf{m}_t)} \mathbf{a}[\mathbf{n}]\delta[\mathbf{m}-\mathbf{n}] \right)^{T} \left(\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}] + \sum_{\mathbf{n} \in \mathcal{N}^{-1}(\mathbf{m}_t)} \mathbf{A}[\mathbf{n}]\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{m}-\mathbf{n}] \right) \\
&\quad - \sum_{\mathbf{m}} \mathbf{e}_{Y_yC_xC_z}{}^{T}[\mathbf{m}]\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}] \\
&= \sum_{\mathbf{n} \in \mathcal{N}^{-1}(\mathbf{m}_t)} \sum_{\mathbf{k} \in \mathcal{N}^{-1}(\mathbf{m}_t)} \mathbf{A}[\mathbf{n}]^{T}\mathbf{A}[\mathbf{k}]\mathbf{c}_{\tilde{p}\tilde{p}}[\mathbf{n}-\mathbf{k}] + \sum_{\mathbf{n} \in \mathcal{N}^{-1}(\mathbf{m}_t)} 2\mathbf{a}[\mathbf{n}]^{T}\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{n}]
\end{aligned} \tag{7.34}$$

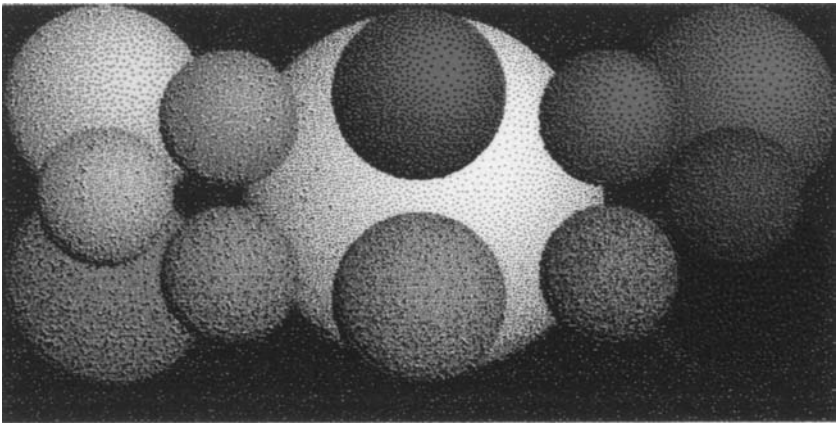
If the toggle under trial is accepted, we modify the $\mathbf{g}^{o}{}_{Y_yC_xC_z}[\mathbf{m}]$ and $\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}]$ LUTs using Equations 7.31 and 7.33, respectively.

Now, consider a trial swap of the halftone pixels $\mathbf{m}_t$ and $\mathbf{m}_s$. This trial change to the halftone will modify the $Y_yC_xC_z$-rendered color image at the pixels in $\mathcal{N}^{-1}(\mathbf{m}_t) \cup \mathcal{N}^{-1}(\mathbf{m}_s)$. Assuming that $\mathbf{m}_t \neq \mathbf{m}_s$, $\mathcal{N}^{-1}(\mathbf{m}_t) \cup \mathcal{N}^{-1}(\mathbf{m}_s)$ may contain either six or seven pixels, depending on the relative locations of $\mathbf{m}_t$ and $\mathbf{m}_s$. Figure 7.10b depicts the seven image pixels that will exhibit a change in $Y_yC_xC_z$ values when the colorant combinations at the two diagonally touching pixels $\mathbf{m}_t$ and $\mathbf{m}_s$ are swapped.

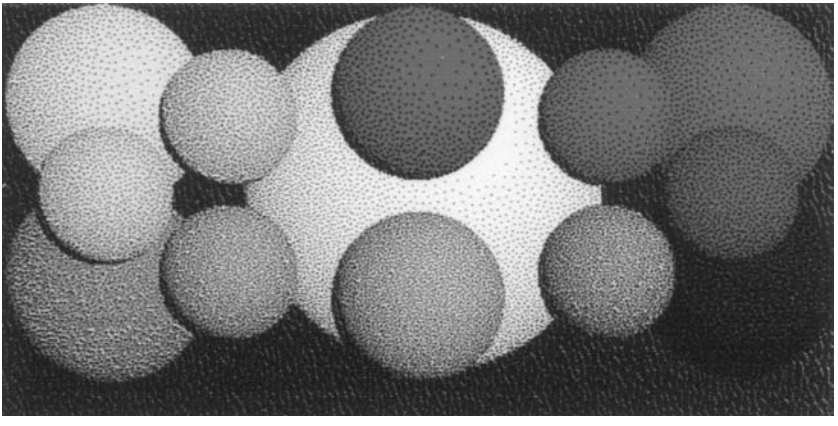
Similar to the trial toggle analysis, we denote the change that this swap will create in the $Y_yC_xC_z$-rendered color image at the pixels $\mathbf{n} \in \mathcal{N}^{-1}(\mathbf{m}_t) \cup \mathcal{N}^{-1}(\mathbf{m}_s)$ by $\mathbf{a}[\mathbf{n}]$. We again define $\mathbf{A}[\mathbf{n}] \equiv \text{diag}(\mathbf{a}[\mathbf{n}])$. The effect of this trial swap on $\mathbf{g}^o{}_{Y_yC_xC_z}[\mathbf{m}]$, $\mathbf{e}_{Y_yC_xC_z}[\mathbf{m}]$, and $\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}]$ is given by the generalization of Equations 7.31 through 7.33 with $\mathcal{N}^{-1}(\mathbf{m}_t)$ replaced by $\mathcal{N}^{-1}(\mathbf{m}_t) \cup \mathcal{N}^{-1}(\mathbf{m}_s)$. Similarly, ΔE_s, the change in E due to this trial swap, is given by Equation 7.34 with the same modification. If the swap under trial is accepted, we alter $\mathbf{g}^o{}_{Y_yC_xC_z}[\mathbf{m}]$ and $\mathbf{c}_{\tilde{p}\tilde{e}}[\mathbf{m}]$ accordingly. We note that, as in the case of regular CDBS, in centering-based CDBS, a swap is a more subtle change (of larger support and smaller effect per pixel) to the $Y_yC_xC_z$-rendered color image than a toggle, and it tends to conserve the average local "color" in the area containing the swapped pixels. We also note that, due to the fact that a toggle in centering-based CDBS brings a much more subtle change to the halftone than one in regular CDBS, we obtain very high-quality halftones doing only toggles in centering-based CDBS. Unlike regular CDBS and monochrome DBS, swaps do not lead to a significant improvement in texture. As with regular CDBS, for efficient implementation of centering-based CDBS, we again exploit the modified search strategies discussed above.

Figure 7.11 shows 600-dpi scans of the 300-dpi halftones of the *faxballs* image obtained using CDBS and centering-based CBDS. Comparison of Figures 7.11a and 7.11b, which were both created using the simple $RGB \rightarrow Y_yC_xC_z$ calibration shows that the 2×2 centering model has been very effective in reducing dot interaction artifacts, especially in the green and cyan balls. The dot interaction model also improves the color textures and gives us much smoother transitions between the light and dark areas. It aims to improve halftone texture and does not try to preserve lightness and saturation. Therefore, Figure 7.11b is generally lighter and less saturated than Figure 7.11a. This side effect may be eliminated with a tone correction prior to halftoning.

Both CDBS (Figure 7.11a) and centering-based CDBS (Figure 7.11b) with the simple $RGB \rightarrow Y_yC_xC_z$ calibration method give colorant-wise acceptable results for the *faxballs* image, but applying CDBS to the ramps of six Neugebauer primaries (*CMYRGB*) (Figure 7.17a) reveals the need for a different $RGB \rightarrow Y_yC_xC_z$ calibration method for business graphics where the intent is to preserve the purity of colors. In Figure 7.17a, we notice *phantom* dots, i.e., dots with wrong colorant values such as magenta dots in the yellow ramp and cyan dots in the red ramp. This is a consequence of the fact that CDBS (centering-based CDBS) operates with an input in the device-inde-



(a)



(b)

Figure 7.11 ***(See color insert)*** 600-dpi scans of the 300-dpi halftones of the *faxballs* images obtained using (a) CDBS and (b) centering-based CDBS with the simple *RGB* $\rightarrow Y_yC_xC_z$ calibration.

pendent space of $Y_yC_xC_z$ and that it is not a colorant-space-based, colorant-level-preserving halftoning algorithm. CDBS minimizes a least-squares error metric in the $Y_yC_xC_z$ space, and this error minimization favors color textures that are pleasing to the human viewer due to the HVS model incorporated. Our experimentation has shown that CDBS is biased to create visually pleasing halftones at the expense of leading to $Y_yC_xC_z$ values that are somewhat different from the desired input $Y_yC_xC_z$ values, especially in the very dark colors. Subject to this bias and limit in accuracy, CDBS is a colorimetric technique developed to minimize a measure of perceived error in $Y_yC_xC_z$. Therefore, it is not calibrated on a colorant basis. This is mainly due to the fact that different spatial arrangements of a single desired colo-

rant combination in a region in the image can lead to different $Y_yC_xC_z$ values. To allow the CDBS algorithm to be used in applications where the calibration of the color printer on a colorant basis is important, such as business graphics as in the example of the ramps of six Neugebauer primaries, we devise the following iterative $RGB \rightarrow Y_yC_xC_z$ calibration scheme that can control the colorant combination in a CDBS halftone.

7.4.4 *Iterative RGB → $Y_yC_xC_z$ calibration*

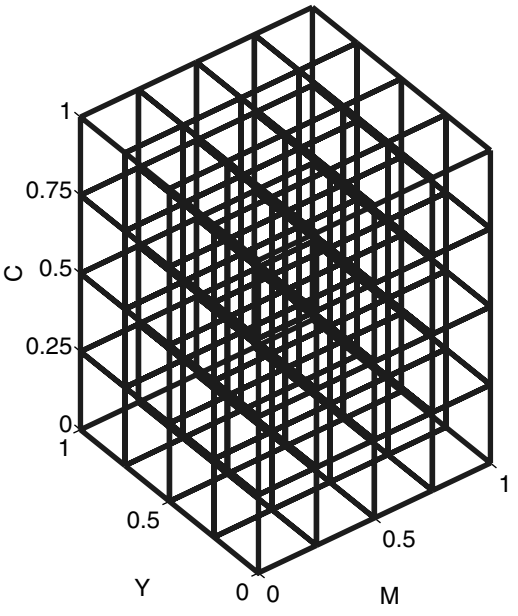
This method uses a cellular measurement-based grid structure that is iteratively updated and then exploits tetrahedral interpolation. It is developed solely for centering-based CDBS. To convert *RGB* $\mathbf{f}[\mathbf{m}]$ to $\mathbf{f}_{Y_yC_xC_z}[\mathbf{m}]$, we first change the *RGB* values to *CMY* using $C = 1 - R$, $M = 1 - G$, $Y = 1 - B$. We denote this continuous pseudo-colorant space by $\overline{CMY}$. Our goal is to increase the accuracy of the simple undercolor-removal-strategy-based *RGB* $\rightarrow Y_yC_xC_z$ calibration method using a cellular-measurement-based grid structure and iterative update of the grid structure. This calibration method does not require extra colorimetric measurements. It uses the 1072 measurements made before. We first discuss how we construct the initial cellular grid structure. Then, we explain how we update this grid structure.

We start by feeding the measured $Y_yC_xC_z$ values for the 1072 2×2 colorant combinations into CDBS to obtain halftone patches for them. We choose to start with these values, as they form a good estimate of what the printer gamut will be for this halftoning algorithm. Because we macroscopically measure the $Y_yC_xC_z$ values of the tiled version of these 2×2 colorant combinations (Figure 7.9), theoretically, these $Y_yC_xC_z$ values correspond to $\overline{CMY}$ values

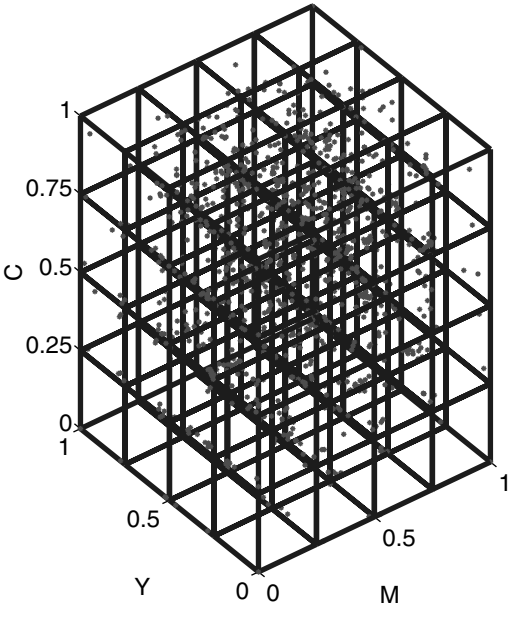
$$\left\{\frac{n_C}{4}, \frac{n_M}{4}, \frac{n_Y}{4}\right\}$$

where $n_C, n_M, n_Y \in \{0, 1, 2, 3, 4\}$. For a four-colorant printer with full undercolor removal, we assume that a *K* dot is equivalent to *C*, *M*, and *Y* dots put down together. These theoretical $\overline{CMY}$ values are the grid points of the uniform $5 \times 5 \times 5$ grid in $\overline{CMY}$ shown in Figure 7.12a. We compute the actual $\overline{CMY}$ values for the 1072 CDBS halftone patches we created by counting the number of dots of each colorant present in the bit maps. If CDBS were an algorithm calibrated on a colorant basis, these values would fill the whole unit cube in $\overline{CMY}$ and would be exactly equal to the grid points of the $5 \times 5 \times 5$ $\overline{CMY}$ grid of Figure 7.12a. We typically obtain $\overline{CMY}$ values as shown in Figure 7.12b.

Among these 1072 $\overline{CMY}$ values, we pick the 125 that are closest in Euclidean distance to the grid points of the $5 \times 5 \times 5$ grid in $\overline{CMY}$. This gives us a nonuniform grid structure that does not fill the unit cube in $\overline{CMY}$ as shown in Figure 7.13a and a corresponding nonuniform grid structure in $Y_yC_xC_z$ as shown Figure 7.13b. There is a 1-1 and onto mapping between the

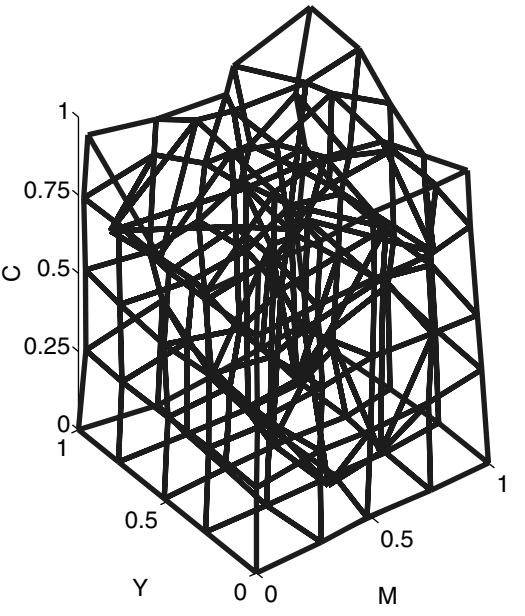


(a)

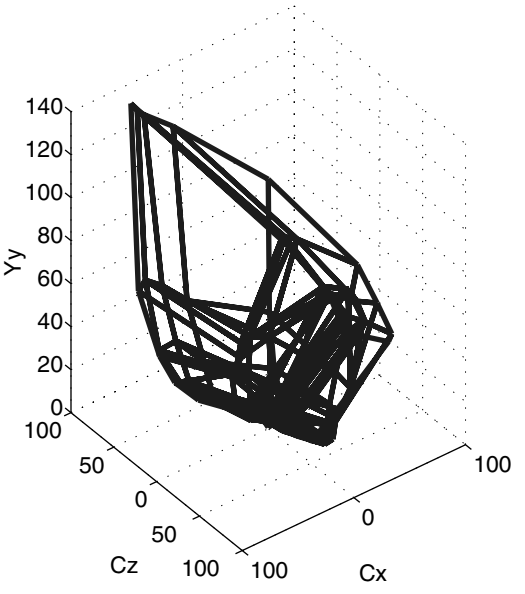


(b)

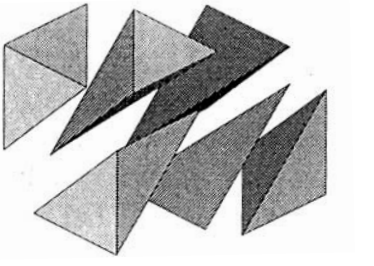
Figure 7.12 (a) Theoretical $5 \times 5 \times 5$ $\overline{CMY}$ grid and (b) actual $\overline{CMY}$ measurements superimposed on the $5 \times 5 \times 5$ grid.



(a)



(b)



(c)

Figure 7.13 (a) Actual $\overline{CMY}$ grid, (b) $Y_yC_xC_z$ grid, and (c) tesselation of a subcube in $\overline{CMY}$ into tetrahedra.

grid points of the ideal $5 \times 5 \times 5$ $\overline{CMY}$ grid, the actual $\overline{CMY}$ grid, and the $Y_y C_x C_z$ grid.

We then divide each subcube of the ideal $5 \times 5 \times 5$ $\overline{CMY}$ grid into six tetrahedra as shown in Figure 7.13c. We use this specific tetrahedralization among all possible tetrahedralizations of a cube, because deviations from the equi-faced tetrahedra are limited, and all interfaces between tetrahedra are oppositely congruent, as stated necessary by Gennetten[65] for accurate interpolation. Furthermore, this tetrahedralization with the lower left vertex located at the smallest CMY value satisfies the minimum brightness variance criterion,[66] which states that, in selecting from possible colorant sets to render a desired color in a halftone, choosing the set with minimum brightness variation reduces the halftone noise and results in visually more pleasing halftones. We carry out the same tessellation in the actual $\overline{CMY}$ grid and the $Y_y C_x C_z$ grid. Figure 7.14 depicts the two-dimensional versions of these three grid structures wherein each subsquare is divided into two triangles. For a desired $\overline{CMY}$, we locate the tetrahedron that the color belongs to in the ideal $\overline{CMY}$ grid and calculate interpolation weights. Then, we use the $Y_y C_x C_z$ values of the vertices of the corresponding tetrahedron in the $Y_y C_x C_z$ space to interpolate the $Y_y C_x C_z$ value for the desired $\overline{CMY}$.

Even though this calibration method gives better results than the simple Neugebauer-model-based $RGB \rightarrow Y_y C_x C_z$ transformation discussed above, it is still incapable of producing CDBS halftones that span the whole unit cube in $\overline{CMY}$, since the rendered colors are restricted to the ones in the convex hull of the nonuniform grid structure shown in Figure 7.13a. We take the following iterative grid expansion and correction approach to fill a larger portion of the $\overline{CMY}$ unit cube, and to bring the grid points closer to the grid points in the ideal $\overline{CMY}$ grid, i.e., to render a larger set of colors and to attain more accurate colorant distributions using CDBS.

For a given grid point CMY_{target} in the ideal $5 \times 5 \times 5$ $\overline{CMY}$ grid, we find the tetrahedra in the ideal $\overline{CMY}$ grid for which this point is a vertex.

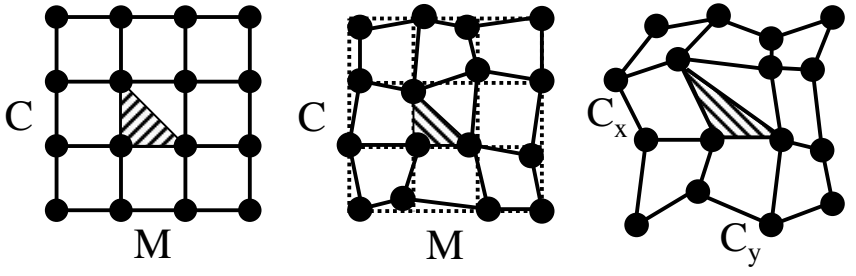


Figure 7.14 Illustration of the relation between different grid structures for a two-dimensional example with a 3×3 grid: (a) ideal $\overline{CMY}$ grid, (b) actual $\overline{CMY}$ grid, and (c) $Y_y C_x C_z$ grid.

Depending on the location of CMY_{target} in the unit cube, CMY_{target} may be a vertex of up to 20 tetrahedra. For each such tetrahedron, we use the corresponding tetrahedron in the actual $\overline{CMY}$ grid to find the weights for tetrahedral extrapolation or interpolation of CMY_{target}. Figure 7.15a depicts a grid point extrapolated from two tetrahedra. Using these weights and the $Y_yC_xC_z$ values of the vertices of the corresponding tetrahedra in the $Y_yC_xC_z$ grid, we find approximate $Y_yC_xC_z$ values for CMY_{target}. We feed these values into CDBS, obtain halftone patches, and compute the $\overline{CMY}$ for the resulting halftones.

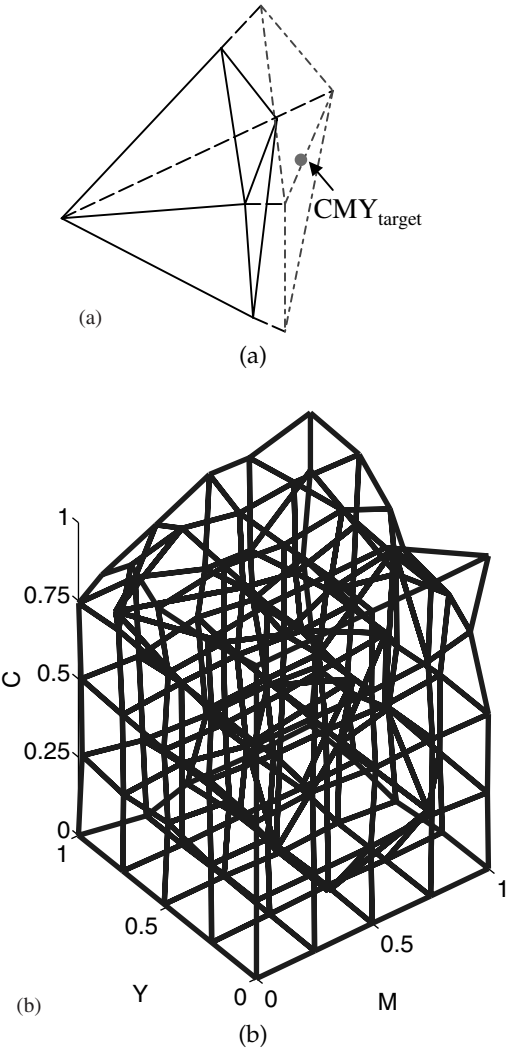


Figure 7.15 (a) Extrapolation of CMY_{target} and (b) the $\overline{CMY}$ grid structure after the first update.

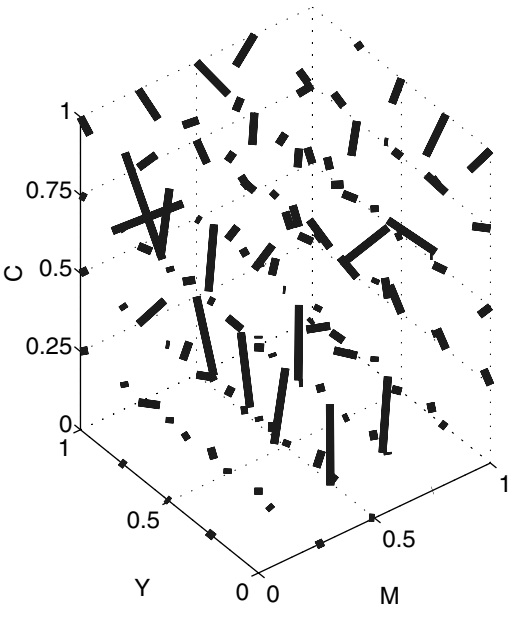
We pick the $\overline{CMY}$ value with minimum Euclidean distance ΔCMY from CMY_{target}. If this new $\overline{CMY}$ value for CMY_{target} is closer to CMY_{target} then the already existing grid point in the actual $\overline{CMY}$ grid structure, we update our $\overline{CMY}$ and $Y_y C_x C_z$ grid structures. We repeat this process in parallel for all of the 125 grid points. This expansion and correction approach may be iterated until the desired accuracy in the *CMY* rendition is obtained with the halftones. Figure 7.15b shows the new iterated $\overline{CMY}$ grid structure that is noticeably closer to the ideal $5 \times 5 \times 5$ grid. The ΔCMY error vectors for the initial $\overline{CMY}$ grid structure (Figure 7.13a) and its iterated version (Figure 7.15b) are depicted in Figure 7.16, and the maximum and mean ΔCMY values are summarized in Table 7.2.

Table 7.2 Comparison of Initial and Iterated $\overline{CMY}$ Grid Structures

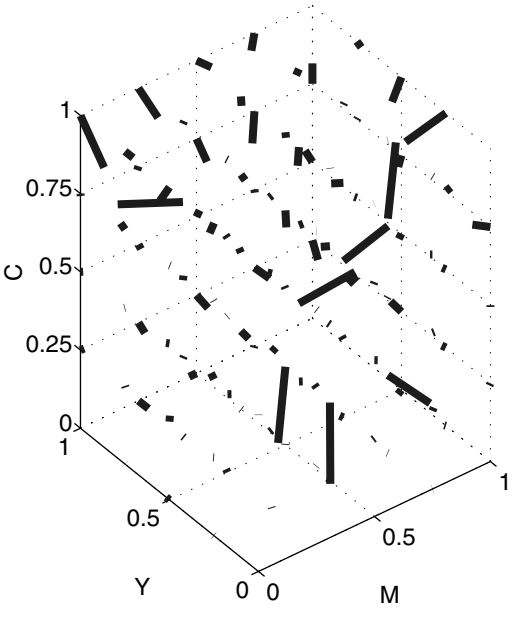
	Initial $\overline{CMY}$ Grid	Iterated $\overline{CMY}$ Grid
Maximum ΔCMY	0.3221	0.2607
Mean ΔCMY	0.0703	0.0383

The expansion and correction approach provides a significant decrease in the ΔCMY errors. Comparison of the scan of the halftone of the ramps of the Neugebauer primaries obtained using centering-based CDBS with this iterated $\overline{CMY}$ grid-structure-based calibration (Figure 7.17b) with that obtained using centering-based CDBS with the simple calibration (Figure 7.17a) emphasizes what we have gained by incorporating the iterative calibration. The colorant rendition is almost perfect, i.e., we have considerably fewer *phantom* dots. The subtractive primary ramps (*C*, *M*, *Y*) are almost solely made up of dots of the desired respective colorant. For example, there are almost no magenta dots in the cyan and the yellow ramps. Similarly, the subtractive secondary ramps (*R*, *G*, *B*) contain very few dots of the wrong colorant; e.g., the number of cyan dots in the red ramp and the number of magenta dots in the green ramp have decreased significantly. While attaining improved colorant rendition, we still maintain the visually pleasing smooth color textures of CDBS.

In the presentation of this halftoning calibration algorithm, we have assumed that the color printer has three colorants *CMY*. However, this algorithm can also be applied to four-colorant *CMYK* printers with full undercolor removal. For a four-colorant *CMYK* printer with full undercolor removal, each *CMYK* value can be mapped to a *CMY* value according to $CMYK \rightarrow (C + K)(M + K)(Y + K)$. Because favoring composite black dots over process black dots improves the color texture of the halftones, we modify the previously described method for choosing the points for our grid structures in $\overline{CMY}$ as follows. We add an empirically determined penalty term $(K - min(CMY_{grid})^2)^2$ if $K > min(CMY_{grid})^2$ to our distance metric ΔCMY. That is, given two candidate points for a given grid point with the same *CMY* values, we favor the one with less *K* coverage. This penalty serves to limit

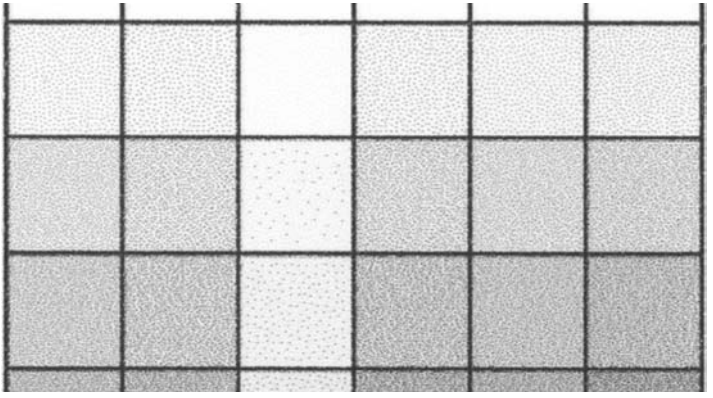


(a)

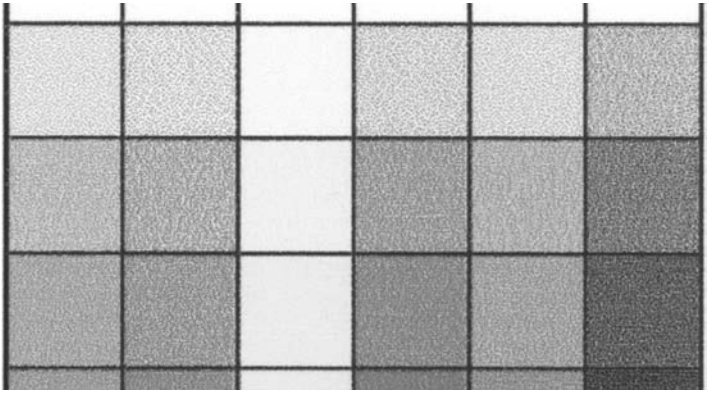


(b)

Figure 7.16 DCMY error vectors of (a) the initial $\overline{CMY}$ grid structure and (b) the iterated $\overline{CMY}$ grid structure.



(a)



(b)

Figure 7.17 ***(See color insert)*** 600-dpi scans of the 300-dpi halftones of the ramps of the six Neugebauer primaries obtained using (a) centering-based CDBS with the simple calibration and (b) centering-based CDBS with the iterated $\overline{CMY}$ grid-structure-based calibration.

the process K in the halftones, like the modification to the interpolation weights discussed in simple undercolor-removal-strategy based $RGB \rightarrow Y_y C_x C_z$ transformation.

7.5 HVS-model-based color error diffusion

Error diffusion has been more successfully applied to color halftoning using the generalized version, vector error diffusion, where the error due to halftoning at a given pixel is diffused to color planes jointly or the error diffusion is carried out in a device independent space. However, vector error diffusion poses unique challenges, as pointed out by Sharma and Trussell.[67] Haneishi

et al.[68] employed vector error diffusion in the device-independent color spaces of *XYZ* and CIE, compared their performances to scalar error diffusion in *RGB,* and proposed a method to decrease the spatial artifacts due to error accumulation. Marcu and Abe[69] investigated the effect of the shape and the size of the threshold array, the weights, and the color space in vector error diffusion. Shu[70] proposed a modified scalar error diffusion that uses the inter-color frame knowledge to enforce exclusion of nonharmonic colors in a local area in the image. Klassen et al.[63] applied vector error diffusion in a distorted color space, where they made it less likely to have dark primaries (*RGBK*) in light regions and light primaries (*WCMY*) in dark regions. Kim et al.[29] diffused the error over 2×2 pixel blocks. Shu and Boyce,[71] Akarun et al.,[72] and Bozkurt et al.[73] used adaptive vector error diffusion wherein they adjusted the diffusion coefficients to improve halftone smoothness or to locally minimize the mean-squared error between desired and rendered colors. Lau et al.[74] combined Knox and Eschbach's threshold modulation[75] and Levien's output-dependent feedback[76] in a vector framework and added an interference matrix to control the overlap of dots from different colorants. Their framework admitted the possibility of coupling between color planes, but they did not provide any guidance regarding how the matrices that controlled this coupling should be designed. Damera-Venkata and Evans[77] propose and validate a model for a vector error diffusion system in which the quantizer is replaced by a gain matrix followed by additive noise that is uncorrelated with the input. They then used this model to design filter coefficients that are optimal under Poirson and Wandell's opponent-color human visual system model.[78]

7.6 HVS-based clustered-dot color screen design

In Section 7.4, we saw how a direct binary search for the optimal color halftone value at each pixel, combined with color rendering device and spatiochromatic HVS models, could effectively develop a very high-quality dispersed dot halftone rendering of a continuous-tone color image. As pointed out earlier, by designing textures for patches of constant color, this approach may also be used as the basis for design of dispersed dot color halftone screens.[49] In Section 7.5, we briefly described the work of Damera-Venkata and Evans,[77] which also employs a numerical search in combination with a spatiochromatic HVS model to optimize the parameters associated with color error diffusion, another dispersed dot halftoning approach. In this section, we again use a color rendering device and spatiochromatic HVS models in combination with a numerical search, but this time, our objective is to optimize the parameters of screens that will generate periodic clustered dot halftones.

In traditional clustered-dot color screening, the screen for each colorant is rotated to a different angle relative to the others. If the angles are not carefully chosen, visible moiré and rosette artifacts may appear. These artifacts primarily result from the interaction of the periodic structures

associated with the halftone screens of different colorants. Registration errors can also introduce unwanted artifacts in the screened images. Using lattice theory and a model for the perceived rendered halftone, we present a systematic method for designing moiré- and rosette-free clustered-dot color screens for discrete-raster color systems. We also investigate strategies for choosing the periodicities so that the resulting screen is robust to registration errors.

Clustered-dot screens produce binary textures in which the individual printer dots are grouped into clusters. To avoid confusion between the individual printer dots and these clusters, we refer to the latter as *macrodots*. When the meaning is clear from the context, we drop the prefix "macro."

The moiré phenomenon is an optical effect that appears when periodic or quasiperiodic structures intersect.[79] These periodic structures could be line gratings or macrodot screens. Moiré consists of a visible pattern that is clearly observed at the intersection, although it is not present in any of the original structures. Unwanted moiré patterns may appear in the printing process for various reasons. For instance, if the image contains some periodic fine details, then a moiré pattern may be caused as an interference between the periodic fine details of the original image and the frequency of the halftone screen. This type of moiré also appears in black-and-white printing. In color printing, the most notorious moiré is due to the superposition of halftone screens of different process colors. A rosette pattern is formed in rotated halftone screens due to the symmetry of the crossline ruling that causes angular separations among screens in the first quadrant to be repeated in the other three quadrants, forming a circular or polygonal pattern.[1] Generally, rosette patterns are relatively less objectionable than moiré patterns, but they represent an unstable moiré free state. Rosette patterns can be clear centered, dot centered, or something in between.

The theory of moiré suppression in monochrome halftoning is well studied. Several techniques have been proposed. For example, Allebach and Liu[80] proposed a random quasiperiodic halftone screen. It consists of a periodic array of cells where, within each cell, the screen profile is independently randomized. The fixed cell structure and independence between cells results in a loss of macrodot and macrohole integrity. A random nucleated screen that maintains the integrity of macrodots and macroholes has also been suggested.[81] Using the nonlinearities in the spectral analysis of the halftone image to suppress moiré has also been investigated.[82,83] Roetling[84] proposed the ARIES method that adjusted the thresholds in the dither matrix cell by cell to ensure correct average tone over the halftone cell, thus suppressing moiré that results from slowly varying periodic errors in average tone. Because ARIES is based on discrete cells, it can introduce artifacts at the cell boundaries. Levien[85] extended ARIES to suppress boundary artifacts. Rao and Arce[86] looked into improving clustered-dot screening by using arbitrary screen periodicities. A frequency domain analysis of screening for arbitrary sampling lattices and screen periodicities has also been done.[87] Green noise

masks that reduce moiré by randomizing the location and size of the dot clusters have also been explored.[74,88]

Mathematical approaches analyzing moiré due to superposition of clustered-dot screens for color halftoning have also been reported. The geometric model[1,89] is based on periods and angles of the superposed layers. It provides equations that under certain limitations predict the geometric properties of the moiré patterns. The algebraic approach[90] also gives the same result. Amidror et al.[91] analyzed moiré formation in the frequency domain, while Kaji et al.[92] proposed a method to minimize moiré at the expense of rosette formation.

It is suggested in the literature[1] that, for a three-colorant halftone, the screen angles should be separated by 30°. The angles have to be quite accurate to avoid the appearance of moiré. In the graphic arts industry, the rotation angles are achieved by using the following three approaches:[93] rational tangent, irrational tangent, and rational supercell. The rational tangent method (conventional approach) only allows integer elements in the periodicity matrices, thereby restricting the available screen angles. The irrational tangent technique[94] is used in high-end graphic arts plotters. It employs floating-point arithmetic and registers to attain precise angles. The drawback is that the screen cells are different from location to location, which can cause moiré.[93] In the rational supercell approach, a large cell that contains an array of cells is fit to the raster grid. Screen angles can be achieved to any desired precision by increasing the supercell size. This method requires less processing than the irrational tangent method. However, the possibility of moiré still exists. To allow more latitude in the choice of screen angles, nonorthogonal halftone screens that avoid moiré have also been proposed.[95] While the theory of moiré and rosette formation due to superposition of the halftone screens has been thoroughly studied, little work has been done to develop a systematic method for designing moiré- and rosette-free clustered-dot color screens. Mostly, the screen angles are chosen based on the designer's experience.

Lattice theory is a well-developed branch of abstract algebra that has various applications in signal and image processing. It has been used in the time sequential sampling problem[96–98] and for the analysis of multidimensional multi-rate filter banks.[99] We use it to analyze the structure of CMYK clustered-dot halftone patterns with arbitrary periodicities and offsets for the individual colorants.

A CMYK halftone screen is completely specified by the periodicity matrix, the offset vector, and the dot profile for each of the four colorants. We use a perceptual model to determine the perceived error between the texture predicted by the halftone and the input image and jointly optimize over the set of periodicities and offsets to minimize this error. The objective is to ascertain periodicity matrices and offsets for the colorants that minimize the visibility of moiré and rosette artifacts. In addition, the screens should be robust to registration errors.

7.6.1 Preliminaries

As we did in Section 7.4, we represent spatial quantities, i.e., position or displacement by boldface lower-case letters. We use **[m]** to represent discrete spatial coordinates and **(x)** to represent continuous spatial coordinates. The function $\mathbf{f}_{type}[\mathbf{m}]$ denotes the three-tuple discrete space continuous-tone image, and $\mathbf{g}_{type}[\mathbf{m}]$ represents the three-tuple digital halftone, where the subscript *type* is one of *cmy*, *XYZ*, or $Y_y C_x C_z$. When we wish to refer to all three of these simultaneously, we will simply eliminate the subscript. Each pixel of $\mathbf{f}_{cmy}[\mathbf{m}]$ contains a three-tuple of colorant values for the primaries, each ranging between 0 and 1. These colorant values represent the fractional area of coverage for each colorant. The pixels of $\mathbf{g}_{cmy}[\mathbf{m}]$ have the three binary values of the colorants that are either 0 (no colorant) or 1 (100% colorant).

7.6.1.1 Lattices

Multidimensional digital signal processing requires sampling on an array of points. For images, these points are arranged on a plane lattice,[100,101] which is a discrete subgroup L of $\mathcal{R}^2$ defined by two linearly independent vectors $\mathbf{v}_1$ and $\mathbf{v}_2$,[102]

$$L = \{m_1\mathbf{v}_1 + m_2\mathbf{v}_2 | ((m_1, m_2) \in \mathcal{Z})\} \tag{7.35}$$

The generating set $(\mathbf{v}_1, \mathbf{v}_2) \in \mathcal{R}^2$ is called the *lattice basis*. If $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2]$, then L may be expressed as $\{\mathbf{Vm}, \forall \mathbf{m} \in \mathcal{Z}^2\}$, where the sampling matrix $\mathbf{V}$ is nonsingular, and the absolute value of its determinant $|\det \mathbf{V}|$ represents the area of any unit cell. For example, the unit cell could be a parallelogram $\mathcal{P} \in \mathcal{R}^2$, defined by the columns of $\mathbf{V}$ or a Voronoi box.[96] For a lattice L, there are many valid sampling matrices. For example, if $\mathbf{E}$ is any unimodular matrix, i.e., $\det(\mathbf{E}) = \pm 1$, then $\mathbf{V}' = \mathbf{VE}$ is another sampling matrix for L. In this chapter, we assume that the continuous-space, continuous-tone image $\mathbf{f}(\mathbf{x})$ has been sampled on a rectangular lattice $\mathbf{X}$ to yield the discrete-space continuous-tone image $\mathbf{f}[\mathbf{m}]$, where $\mathbf{X} = \text{diag}[X, X]$ and X is the horizontal and vertical distance between printer addressable points. The digital halftone $\mathbf{g}[\mathbf{m}]$ is defined on the same lattice.

7.6.1.2 Periodicity matrix

The two-dimensional discrete-parameter signal $\sigma[\mathbf{m}]$ is said to be periodic if $\exists$ two linearly independent vectors $\mathbf{n}_1$ and $\mathbf{n}_2$ such that $\sigma[\mathbf{m}] = \sigma[\mathbf{m} + \mathbf{Nq}]$, for $\mathbf{N} = [\mathbf{n}_1 | \mathbf{n}_2]$, and $\forall \mathbf{q} \in \mathcal{Z}^2$. The integer matrix $\mathbf{N}$ is called the periodicity matrix. The columns of $\mathbf{N}$ indicate vectors along which the sequence is periodically replicated. $\mathbf{N}$ is nonsingular, and its density $|\det(\mathbf{N})|$ represents the area of a cell that can be replicated to cover the entire two-dimensional space without gaps or overlaps. Like the sampling matrix, for a particular density, the choice of the periodicity matrix is not unique, i.e., $\mathbf{N} = \mathbf{NE}$, where $\mathbf{E}$ is unimodular and is also a valid periodicity matrix. The density of $\mathbf{N}$ gives the number of unique samples contained in a single period.

7.6.2 Clustered-dot color screen design

Each colorant plane is independently halftoned as if it were a separate monochrome image. Screening for the jth colorant consists of thresholding the discrete space signal $\mathbf{f}_j[\mathbf{m}]$ with a spatially periodic threshold matrix $\sigma_j[\mathbf{m}]$ whose periodicity is defined by a periodicity matrix $\mathbf{N}_j$. Because $\sigma_j[\mathbf{m}]$ is periodic with period $\mathbf{N}_j$, it contains $|\det(\mathbf{N}_j)|$ distinct threshold elements $(k + 1/2)/|\det(N_j)|$, $k = 0, 1, \ldots |\det(N_j)| - 1$. If $\mathbf{f}_j$ has constant value b, $(k + 1/2)/|\det(N_j)| \leq b < (k + 2/3)/|\det(N_j)|$, exactly k of the thresholds in each period of σ_j will be exceeded. This will result in the printing of k dots in each period according to the dot profile function[82]

$$\mathbf{g}_j^k[\mathbf{m}] = \begin{cases} 1, & \dfrac{k}{|\det(N_j)|} \geq \delta_j[\mathbf{m}] \\ 0, & \text{otherwise} \end{cases} \tag{7.36}$$

where $\mathbf{g}_j^k[\mathbf{m}] = \mathbf{g}_j^k[\mathbf{m} + \mathbf{N}_j\mathbf{q}]$, for any $\mathbf{q} \in \mathcal{Z}^2$. For the halftone process to be realized by screening, the dot profile function must satisfy the stacking constraint[82]

$$\mathbf{g}_j^l[\mathbf{m}] = 1 \Rightarrow \mathbf{g}_j^k[\mathbf{m}] = 1 \ \forall k \geq l \tag{7.37}$$

As pointed out earlier, the halftone screens for each colorant are also offset relative to each other. Thus, we can write the shifted dot profile function as

$$\mathbf{g}_j^k[\mathbf{m};\mathbf{N}_j, \mathbf{o}_j] = {}^0\mathbf{g}_j^k[\mathbf{m} + \mathbf{o}_j] \tag{7.38}$$

Here, ${}^0\mathbf{g}_j[\mathbf{m} + \mathbf{o}_j]$ is the unshifted dot profile function, and $\mathbf{N}_j$ and $\mathbf{o}_j$ represent the periodicity matrix and the offset vector for the jth colorant, respectively.

For a CMYK halftone screen, we have four periodicity matrices and four offset vectors. For simplicity, here we shall confine our screen design to the cyan, magenta, and yellow colorants. The resulting framework easily can be extended to incorporate the black colorant. The periodicity matrices and offset vectors for the cyan, magenta, and yellow colorants can be expressed as

$$\mathbf{N}_c = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}, \mathbf{N}_m = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}, \mathbf{N}_y = \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix} \tag{7.39}$$

and

$$\mathbf{o}_c = \begin{bmatrix} c_x \\ c_y \end{bmatrix}, \mathbf{o}_m = \begin{bmatrix} m_x \\ m_y \end{bmatrix}, \mathbf{o}_y = \begin{bmatrix} y_x \\ y_y \end{bmatrix} \tag{7.40}$$

Once $\mathbf{N}_c$, $\mathbf{N}_m$, $\mathbf{N}_y$, $\mathbf{o}_c$, $\mathbf{o}_m$, and $\mathbf{o}_y$ are selected, the screen design consists of finding the *discrete parameter halftone cell (DPHC)* and defining the dot profile for each colorant. For ease of notation, we define the following augmented matrices to represent the periodicities and offsets of the cyan, magenta, and yellow colorants:

$$\mathbf{N}_{cmy} = [\mathbf{N}_c|\mathbf{N}_m|\mathbf{N}_y] \tag{7.41}$$

$$\mathbf{o}_{cmy} = [\mathbf{o}_c|\mathbf{o}_m|\mathbf{o}_y] \tag{7.42}$$

In the following, we elucidate the screen design procedure for a given colorant in terms of its periodicity matrix and offset vector. The screens for the other colorants can be similarly designed. The selection of $\mathbf{N}_{cmy}$ and $\mathbf{o}_{cmy}$ will be addressed in Sections 7.6.3 and 7.6.4.

7.6.2.1 Discrete parameter halftone cell

For the purpose of illustration, let the periodicity matrix for a given colorant be

$$\mathbf{N} = [\mathbf{n}_1|\mathbf{n}_2] = \begin{bmatrix} n_{11} & n_{12} \\ n_{21} & n_{22} \end{bmatrix} = \begin{bmatrix} 5 & 2 \\ -2 & 3 \end{bmatrix} \tag{7.43}$$

Without loss of generality, we assume that the offset vector for this colorant is the null vector. We start with a lattice defined by $\mathbf{n}_1$ and $\mathbf{n}_2$. We call the unit cell of this lattice the *continuous parameter halftone cell (CPHC)*. An illustration of the CPHC is given in Figure 7.18. The shaded region represents the *unit cell* $\mathcal{P} \in \mathcal{R}^2$ defined by the columns of $\mathbf{N}$ as

$$\mathcal{P} = \{x\mathbf{n}_1 + y\mathbf{n}_2 | x, y \in \mathcal{R}; (0 \le x, y \le 1)\} \tag{7.44}$$

From the CPHC, we need to find the DPHC. The DPHC should be able to tile the entire two-dimensional discrete-parameter space without gaps or overlaps. To do this, we first compute the number of pixels in the unit cell. This is given by $|\det(\mathbf{N})|$. Then we find the intersecting area of the CPHC and the pixels in the DPHC by using the procedure outlined in Reference 103. Generally, computing the overlapping area of two arbitrary polygons will have quadratic complexity in the number of vertices of the polygons. However, because the CPHC is always convex, the overlapping area can be determined with linear complexity.[103] The area of overlap of each pixel for the CPHC given by the periodicity matrix in Equation 7.43 is shown in Figure 7.19a. Here, 0 indicates that the pixel does not overlap with the CPHC, and 1 means that the pixel is completely contained in the CPHC.

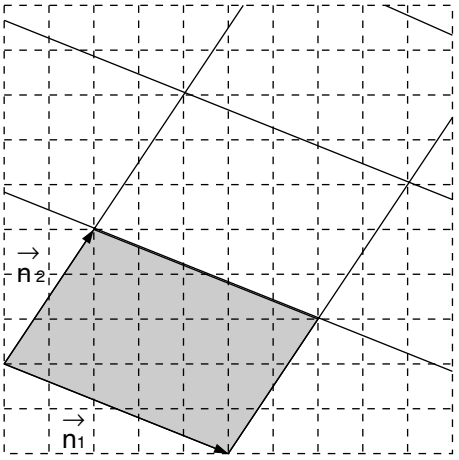


Figure 7.18 Continuous parameter halftone cell defined by $\mathbf{N} = [\mathbf{n}_1, \mathbf{n}_2] = [(5\ -2)^T; (2\ 3)^T]$.

(a)

0	0.33	0.8	0.4	0.05	0	0
0.08	0.92	1	1	0.95	0.6	0.2
0.67	1	1	1	1	1	0.67
0.2	0.6	0.95	1	1	0.92	0.08
0	0	0.05	0.4	0.8	0.33	

(b)

		X				
	X	X	X	X	X	
X	X	X	X	X	X	X
	X	X	X	X	X	
				X		

Figure 7.19 Finding the discrete parameter halftone cell for $\mathbf{N} = [(5\ -2)^T; (2\ 3)^T]$, $\det(\mathbf{N}) = 19$: (a) area of overlap of each pixel with the continuous parameter halftone cell and (b) discrete parameter halftone cell.

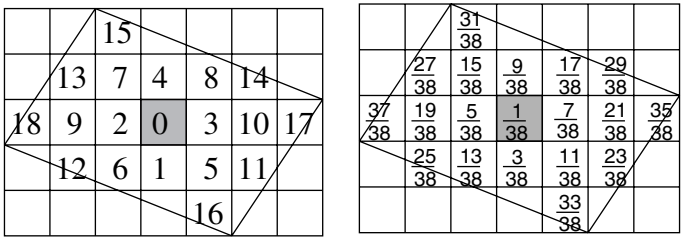
Then we assign pixels to the unit cell in order of decreasing area of overlap with the CPHC. During this process, we skip the pixels that are congruent to a pixel that has already been assigned to the DPHC. To determine if the pixel at location $\mathbf{l}$ is congruent to the pixel at location $\mathbf{m}$, we compute $\mathbf{q} = \mathbf{N}^{-1}[\mathbf{m} - \mathbf{l}]$ and test to see if $\mathbf{q} \in Z^2$. Note that because $\mathbf{N}$ has linearly independent columns, it is always invertible. The pixels assigned to the DPHC are marked by the letter X in Figure 7.19b.

7.6.2.2 *Macrodot shape and growth*

The best macrodot/macrohole shape is generally accepted to be circular in the highlights and shadows, becoming elongated as we move toward the mid-tones so that dots/holes join at two points rather than four.[104] For simplicity, we used a fixed circular dot/hole shape throughout the tone scale, although our algorithm can easily handle the more general case where the dot/hole shape varies with gray level.

Starting from the center of the DPHC, we grow the macrodot in a contiguous manner such that pixels having smaller Euclidean distance from the center are assigned first. In this manner, we define an index matrix $i[\mathbf{m}]$. It takes on integer values 0, 1, … $|\det(\mathbf{N})| - 1$ that indicate the order in which the dots are added to the DPHC. Specifying the index matrix in this manner results in a dot profile that satisfies the stacking constraint. Finally, the threshold matrix $\sigma[\mathbf{m}] = (i[\mathbf{m}] + 0.5)/|\det(\mathbf{N})|$ is computed. The sequence of index values and the corresponding thresholds for the DPHC in Figure 7.19 are given in Figure 7.20.

The textures for various absorptances for the threshold matrix in Figure 7.20b are shown in Figure 7.21. The macrodots develop as expected in the highlights (Figure 7.21a) and mid-tones (Figure 7.21b). However, in the shadow areas (Figure 7.21c), the macroholes do not have a circular structure and can give rise to objectionable textures. Macrodots grow by starting with the smallest threshold and increasing to the largest threshold. To preserve the integrity of both macrodots and macroholes, we follow the procedure described in Reference 81, i.e., grow holes (simultaneously with dots) in the 3×3 neighborhood of halftone cells by starting with the largest threshold



(a) (b)

Figure 7.20 (a) Index values and (b) thresholds for the DPHC in Figure 7.19.

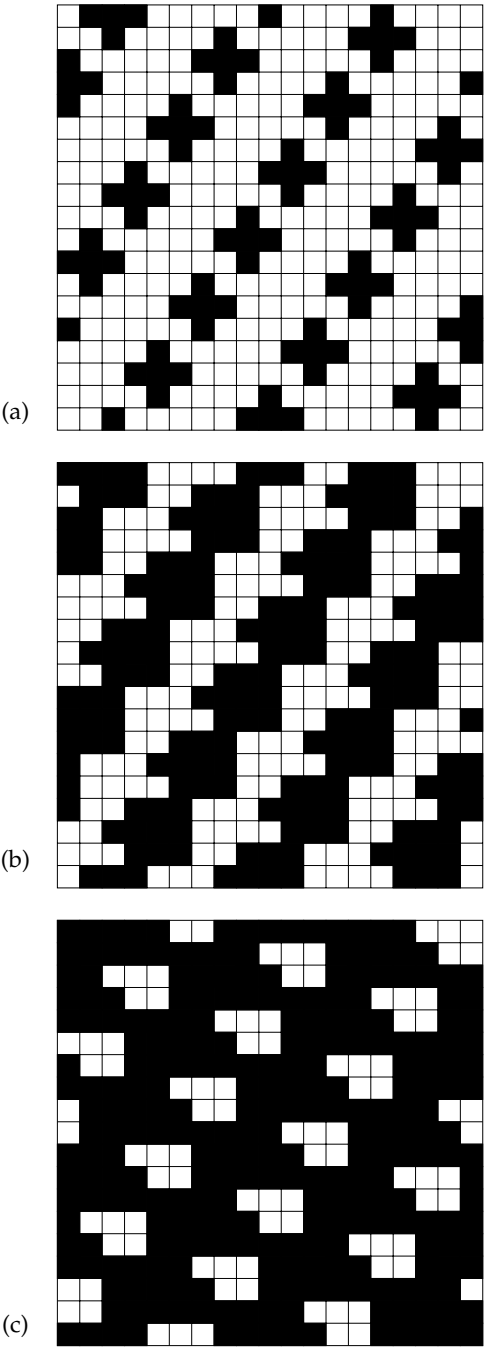


Figure 7.21 Textures obtained by growing dots only according to the threshold matrix in Figure 7.20b: (a) absorptance = 0.26, (b) absorptance = 0.53, and (c) absorptance = 0.74.

and decreasing to the lowest threshold. Without loss of generality, we assume that the leftmost vertex is located at the origin. The macrodot centers are at the center of the cells, while the macrohole centers are positioned at the vertices. The location of the 9 macrodot and 16 macrohole centers can be written in terms of the periodicity matrix $\mathbf{N} = [\mathbf{n}_1 | \mathbf{n}_2]$ as

$$\mathbf{c}_d^{\mathbf{m}} = \left\lceil \frac{1}{2}(\mathbf{n}_1 + \mathbf{n}_2) \right\rfloor + \mathbf{N}\mathbf{m}, \qquad \mathbf{m} \in [0, 1, 2]^2$$

$$\mathbf{c}_h^{\mathbf{m}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \mathbf{N}\mathbf{m}, \qquad \mathbf{m} \in [0, 1, 2, 3]^2$$

where $\lceil \cdot \rfloor$ indicates rounding to the nearest integer, $[0, 1, 2]^2$ denotes the nine-point set of two-tuples formed by the Cartesian product of [0, 1, 2] with itself, and $[0, 1, 2, 3]^2$ is defined similarly.

Following the sequence of index values shown in Figure 7.20, we grow the 9 dots and 16 holes simultaneously. If a location has already been taken, we skip to the next available position, proceeding in order of increasing distance from the dot/hole center. In this way, the circular structure of both dots and holes is preserved. As an example, consider growing only one dot and two holes as shown in Figure 7.22. When we get to the ninth entry for the hole to the right of the dot, this location is already taken by the sixth entry for the dot. So, we go to the next available location. The resulting thresholds for our example are shown in Figure 7.23. The textures obtained by growing dots and holes simultaneously are given in Figure 7.24. Note

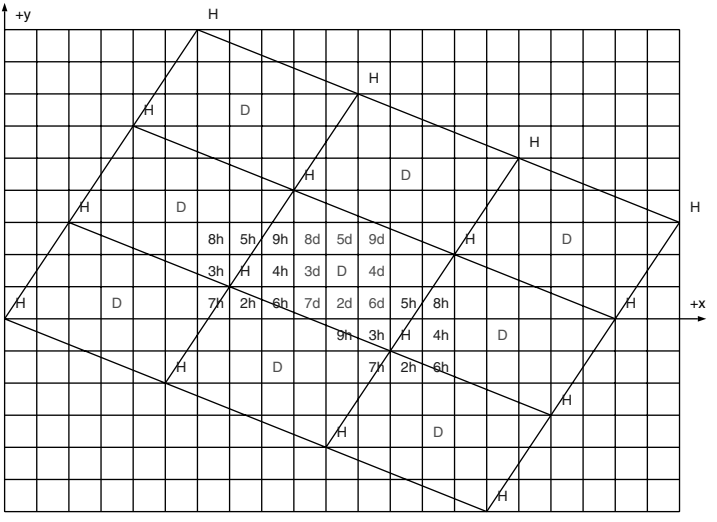


Figure 7.22 Illustration of growing dots and holes simultaneously.

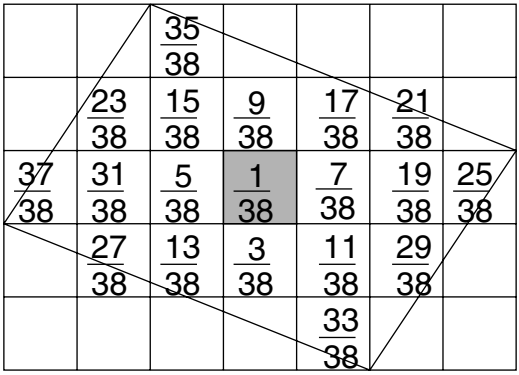


Figure 7.23 Threshold matrix obtained by growing dots and holes simultaneously.

that holes retain the circular structure in the shadow areas. Now we have a way of generating the halftone cell for a periodicity matrix of arbitrary shape. To facilitate implementation, we represent it by a rectangular region.

7.6.2.3 *Representing a nonrectangular halftone cell by a rectangular region*

An equivalent rectangular halftone cell of the same density can be created from the nonrectangular halftone cell by using the method described by Holladay.[105] In this method, the periodicity matrix **N** is used to obtain four parameters.

1. Height K of the equivalent rectangular region
2. Width L of the equivalent rectangular region
3. Shift D in the vertical direction
4. Shift E in the horizontal direction

Successive displacements by D in the vertical direction and E in the horizontal direction will translate the original sub-element of the halftone cell to the next corresponding position in the vertical direction. For the periodicity matrix in Equation 7.43, $K = L = 19$, $D = 1$, and $E = 7$. These variables are shown in Figure 7.25. Here, the shaded region corresponds to the equivalent rectangular region of the halftone cell.

Using the method described in this section, we can generate the halftone $g^{k}_{cmy}[\mathbf{m}; \mathbf{N}_{cmy}, \mathbf{o}_{cmy}]$ for arbitrary periodicities and offsets for any absorptance level k. In the following section, we describe the perceptual model and the error metrics used for selecting $\mathbf{N}_{cmy}$ and $\mathbf{o}_{cmy}$.

7.6.3 *Printer and perceptual model and error metrics*

The objective is to find periodicity matrices and offset vectors that achieve a uniform rendition of any printable color using cyan, magenta, and yellow

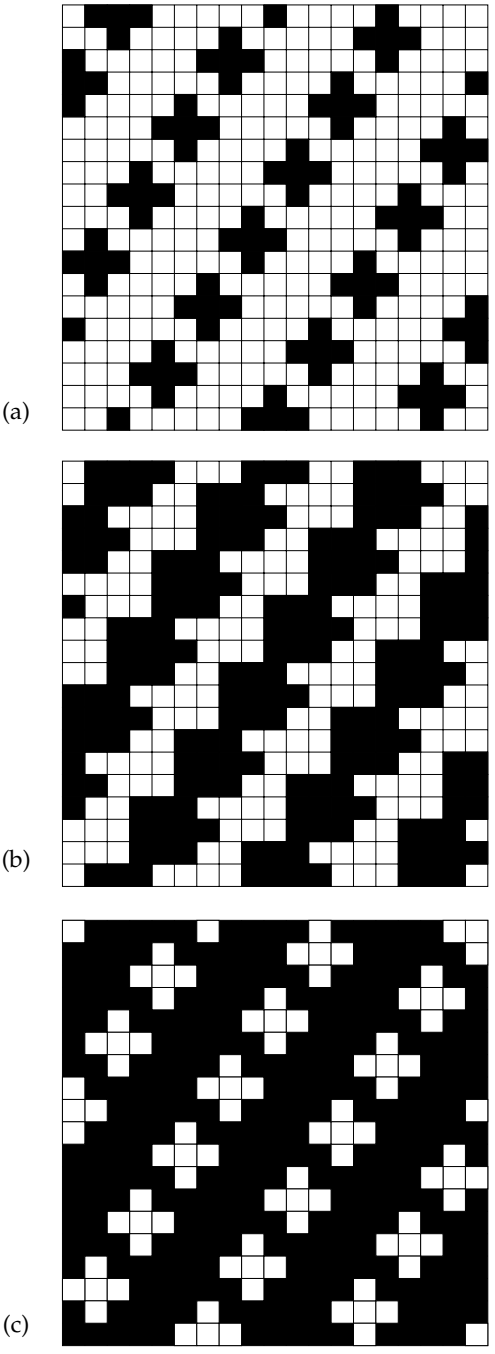


Figure 7.24 Textures obtained by growing dots and holes simultaneously for the index values in Figure 7.23: (a) absorptance = 0.26, (b) absorptance = 0.53, and (c) absorptance = 0.74.

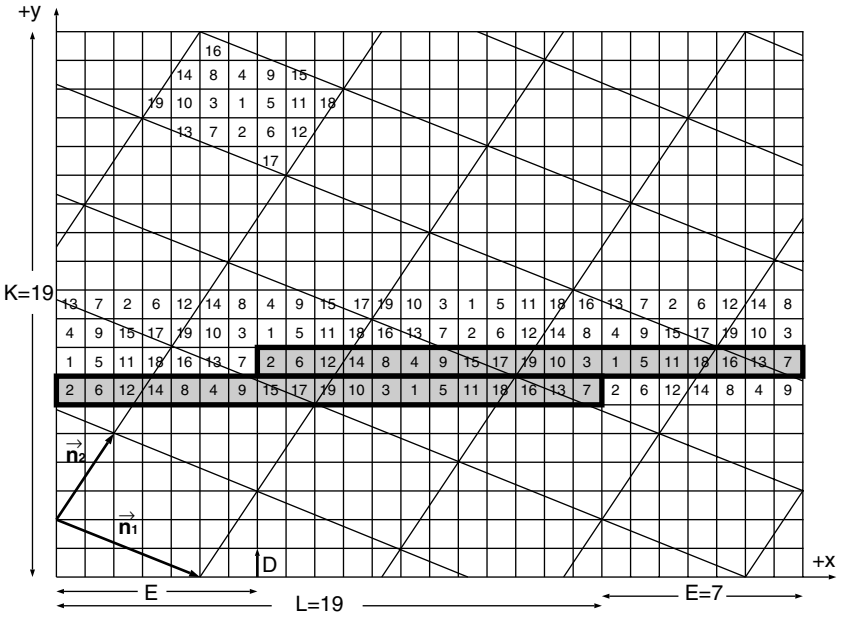


Figure 7.25 Representing a nonrectangular halftone cell by a rectangular region.

colorants. To achieve this, we would like to independently design the halftone texture for all combinations of cyan, magenta, and yellow colorants, where each is present in an amount between 0 and 100% — but all we can do is design the individual textures of the colorants. To achieve our goal, we design halftone patterns for the colorants simultaneously along the neutral axis with each color present in an equal amount. A similar strategy was followed by Lin and Allebach[49] in FM color screen design, but they designed the levels in sequence, so the optimality was conditional on the constraints imposed by the previous level. Here, there is no such conditional dependence. We find the jointly optimal set of periodicity matrices and offset vectors for all textures along the neutral axis.

The perceptual model consists of a color device model (to be discussed shortly) and two other components that were discussed in Section 7.2. These are the linearized uniform color space ($Y_y C_x C_z$) defined in Equation 7.6 and the luminance and chrominance spatial frequency responses described by Equations 7.7 and 7.8, respectively.

7.6.3.1 *Color device model*

With a clustered-dot screen, the resulting macrodots for a given colorant will consist of relatively large regularly shaped areas of contiguous printer dots of the same colorant. Away from the boundary of these colorant areas, the colorant spectral reflectance distribution will match that measured from a

large solid area printed with this colorant. On the shoulder of the macrodot, the spectral distribution will consist of some mixture of that corresponding to the solid colorant and that corresponding to the unprinted substrate. But, because the ratio of macrodot area to macrodot perimeter is large, the contribution of these dot edges can, to a first approximation, be ignored. Thus, we assume a hard, binary macrodot profile in reflectance amplitude.

Similarly, we assume that the spatial microstructure of the edge profile, which is governed by the spatial shape of the individual printer dots, is inconsequential. The validity of this assumption rests with the fact, discussed earlier, that the extent of the point spread function for the human visual system is much greater than the printer pitch **X** and, hence, the size of the individual printer dots. Thus, we simply assume a square $X \times X$ shape for these dots.

Finally, we extend this model to the case in which clustered-dot textures for two or three different colorants overlap. Here we find that the resulting composite print contains relatively large irregularly shaped areas consisting of contiguous printer dots with the same two or three colorants. The spectral distribution of these areas will again match that measured from a large solid area printed with these colorants.

It should be noted that these assumptions can be relaxed by using a table-based equivalent grayscale model,[64] preferably with an offset between the printer and model lattices to reduce the number of measurements required.[36] These steps would be critical for colorimetrically accurate modeling of a dispersed-dot halftoning process.[47] Even these steps will not, however, account for the lateral scattering of light within the substrate from unprinted areas to printed areas where it is trapped. This phenomenon is the Yule Nielsen effect,[18] which was discussed in Section 7.2.

Based on the assumptions discussed above, at any given location **m**, we can print one of the eight possible Neugebauer primaries discussed in Section 7.2, with spectral distributions $R_i(\lambda)$. The measured spectral distributions of the eight Neugebauer primaries printed with an HP DeskJet 970C printer are shown in Figure 7.2. Now that we have discussed the perceptual model, we are in a position to derive an expression for the mean squared error (MSE) and to discuss the metrics used for error minimization.

7.6.3.2 Error metrics

Suppose we have a given set of screens for CMY defined by $(\mathbf{N}_{cmy}, \mathbf{o}_{cmy})$. Assuming that the periodicity matrices of the three colorants have the same fixed density, i.e., $\mathcal{M} = |\det(\mathbf{N}_i)|$, i = *c*, *m*, *y*, we consider the halftone pattern generated for the *k*th level of each of the three screens, where $0 \le k \le \mathcal{M}$. To get the predicted color for the halftone at the *k*th level, we transform the three-tuple halftone $\mathbf{g}^k_{cmy}[\mathbf{m}; \mathbf{N}_{cmy}, \mathbf{o}_{cmy}]$ obtained for the *k*th level to the CIE XYZ color space according to

$$\mathbf{g}^k_{XYZ}[\mathbf{m}; \mathbf{N}_{cmy}, \mathbf{o}_{cmy}] = AIR \tag{7.45}$$

where A is the (3×31) operator matrix whose rows are the color matching functions (CMFs) $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$.[106] I is the (31×31) diagonal matrix containing the relative irradiance spectrum of the D65 illuminant;[106] and R is the 31-element vector containing the measured spectral reflectance of the colorant combination given by $\mathbf{g}^k_{cmy}[\mathbf{m}; \mathbf{N}_{cmy}, \mathbf{o}_{cmy}]$.

So the error image at the kth level can be written as

$$\boldsymbol{\varepsilon}^k_{XYZ}[\mathbf{m}; \mathbf{N}_{cmy}, \mathbf{o}_{cmy}] = \mathbf{g}^k_{XYZ}[\mathbf{m}; \mathbf{N}_{cmy}, \mathbf{o}_{cmy}] - \mathbf{f}^k_{XYZ} \tag{7.46}$$

where $\mathbf{f}^k_{XYZ}$ is the desired color at the kth level obtained by converting fractional CMY colorant amounts to the corresponding CIE XYZ tristimulus value. Using Equation 7.6, this error is transformed to the error $\boldsymbol{\varepsilon}^k_{Y_yC_xC_z}[\mathbf{m}; \mathbf{N}_{cmy}, \mathbf{o}_{cmy}]$ in the linearized uniform color space.

To avoid computing the spatial domain convolution of $\boldsymbol{\varepsilon}^k_{Y_yC_xC_z}[\mathbf{m}; \mathbf{N}_{cmy}, \mathbf{o}_{cmy}]$ and the point spread functions of the HVS corresponding to the luminance and chrominance channels, we evaluate the perceptual error in the Fourier domain. We use the fast Fourier transform (FFT) to efficiently compute the discrete Fourier transforms (DFTs) of the luminance and chrominance channels of $\boldsymbol{\varepsilon}^k_{Y_yC_xC_z}$. To avoid spatial aliasing, the size S of the $S \times S$ colorant bitmaps should be chosen so that it contains an integer number of replications of each of the three periodicities. Also, note that the two-dimensional Fourier transform of a $P \times Q$ matrix typically requires $0.5PQ\log_2(PQ)$ multiplications.[100] So it is desirable to keep the size of the FFT as small as possible.

To determine the smallest size S of the colorant bitmaps that does not result in aliasing, we use the widths of the equivalent rectangular region for the three colorants obtained in Section 7.6.2.3. The largest width $\max(L_c, L_m, L_y)$ of the equivalent rectangular regions of the three colorants gives the smallest size S of the bitmaps that does not result in aliasing. As an illustration, consider the following periodicity matrices:

$$\mathbf{N}_c = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}, \mathbf{N}_m = \begin{bmatrix} 4 & -2 \\ 4 & 2 \end{bmatrix}, \mathbf{N}_y = \begin{bmatrix} 5 & -1 \\ 1 & 3 \end{bmatrix}$$

The widths of the equivalent rectangular regions for the three colorants are $L_c = 4$, $L_m = 8$, and $L_y = 16$. So, for this example, the smallest size of the colorant bitmaps that does not result in aliasing is 16×16. Here, the cyan, magenta, and yellow bitmaps contain 4×4, 2×8, and 1×16 replications of their respective periodicities.

Taking the DFT[100] of $\boldsymbol{\varepsilon}^k_{Y_yC_xC_z}[\mathbf{m}; \mathbf{N}_{cmy}, \mathbf{o}_{cmy}]$, we get

$$\boldsymbol{\varepsilon}^k_i[\mathbf{q}; \mathbf{N}_{cmy}, \mathbf{o}_{cmy}] = \sum_{\mathbf{m} \in \mathbf{R}_\mathbf{S}} \boldsymbol{\varepsilon}^k_i[\mathbf{m}; \mathbf{N}_{cmy}, \mathbf{o}_{cmy}] e^{-j\mathbf{q}^T(2\pi\mathbf{S}^{-1})\mathbf{m}} \tag{7.47}$$

where $i = Y_y, C_x, C_z$
$\mathbf{R}_s$ = region of support
$\mathbf{S} = \text{diag}[S, S]$

As a measure of the error over the entire halftone pattern we use the standard two-norm ($|\,|.|\,|_2$). Therefore, the MSE ΔE_k^2 per pixel at the kth level can be expressed as

$$\begin{aligned}\Delta E_k^2(\mathbf{N}_{cmy}, \mathbf{o}_{cmy}) = \frac{1}{S^2} \sum_{\mathbf{q} \in \mathbf{R}_S} \{&(\kappa \varepsilon_{Y_y}^k[\mathbf{q}; \mathbf{N}_{cmy}, \mathbf{o}_{cmy}] H_{Y_y}[\mathbf{q}])^2 \\ &+ (\varepsilon_{C_x}^k[\mathbf{q}; \mathbf{N}_{cmy}, \mathbf{o}_{cmy}] H_{C_x}[\mathbf{q}])^2 \\ &+ (\varepsilon_{C_z}^k[\mathbf{q}; \mathbf{N}_{cmy}, \mathbf{o}_{cmy}] H_{C_z}[\mathbf{q}])^2\}\end{aligned} \tag{7.48}$$

where $H_{Y_y}[\mathbf{q}] = H_{Y_y}[\mathbf{S}^{-1}\mathbf{q}]$, $H_{C_x, C_z}[\mathbf{q}] = H_{C_x, C_z}[\mathbf{S}^{-1}\mathbf{q}]$, and κ is a weighting factor for adjusting the relative weight between the filtered luminance and chrominance responses, which was discussed in Section 7.2.

To design the screen, we define two error metrics. Given a set of periodicity matrices and offset vectors, these metrics are the average RMSE ϕ_{avg} of all levels and the maximum RMSE ϕ_{max} at any level. They can be expressed as

$$\begin{aligned}\phi_{avg}(\mathbf{N}_{cmy}, \mathbf{o}_{cmy}) &= \frac{1}{\mathcal{M}} \sum_{0 \le k \le \mathcal{M}-1} \Delta E_k(\mathbf{N}_{cmy}, \mathbf{o}_{cmy}) \\ \phi_{max}(\mathbf{N}_{cmy}, \mathbf{o}_{cmy}) &= \max_{0 \le k \le \mathcal{M}-1} \Delta E_k(\mathbf{N}_{cmy}, \mathbf{o}_{cmy})\end{aligned} \tag{7.49}$$

7.6.4 Optimization

To evaluate the efficacy of our technique, we devise the following four optimization conditions for each of the error metrics in Equation 7.49:

1. Jointly minimize the metric over $\mathbf{N}_{cmy}$ and $\mathbf{o}_{cmy}$ to find the best case.
2. Jointly maximize the metric over $\mathbf{N}_{cmy}$ and $\mathbf{o}_{cmy}$ to find the worst case.
3. To evaluate the effect of registration errors for periodicity matrices designed for perfect registration, we use $\mathbf{N}_{cmy}$ obtained for the best case and determine $\mathbf{o}_{cmy}$ that maximizes the error metric.
4. To design screens that are robust to registration errors, we maximize the metric over $\mathbf{o}_{cmy}$ and then minimize over $\mathbf{N}_{cmy}$.

The eight conditions (four for each metric) are listed in Table 7.3.

Table 7.3 Optimization Conditions and Their Interpretation

Case	Expression	Interpretation
1	$(\mathbf{N}^1_{cmy}, \mathbf{o}^1_{cmy}) = \arg \min_{\mathbf{N}_{cmy}, \mathbf{o}_{cmy}} \{\phi_{avg}(\mathbf{N}_{cmy}, \mathbf{o}_{cmy})\}$	minimize ϕ_{avg} over $\mathbf{N}_{cmy}$ and $\mathbf{o}_{cmy}$
2	$(\mathbf{N}^2_{cmy}, \mathbf{o}^2_{cmy}) = \arg \min_{\mathbf{N}_{cmy}, \mathbf{o}_{cmy}} \{\phi_{max}(\mathbf{N}_{cmy}, \mathbf{o}_{cmy})\}$	minimize ϕ_{max} over $\mathbf{N}_{cmy}$ and $\mathbf{o}_{cmy}$
3	$(\mathbf{N}^3_{cmy}, \mathbf{o}^3_{cmy}) = \arg \max_{\mathbf{N}_{cmy}, \mathbf{o}_{cmy}} \{\phi_{avg}(\mathbf{N}_{cmy}, \mathbf{o}_{cmy})\}$	maximize ϕ_{avg} over $\mathbf{N}_{cmy}$ and $\mathbf{o}_{cmy}$
4	$(\mathbf{N}^4_{cmy}, \mathbf{o}^4_{cmy}) = \arg \max_{\mathbf{N}_{cmy}, \mathbf{o}_{cmy}} \{\phi_{max}(\mathbf{N}_{cmy}, \mathbf{o}_{cmy})\}$	maximize ϕ_{max} over $\mathbf{N}_{cmy}$ and $\mathbf{o}_{cmy}$
5	$(\mathbf{N}^1_{cmy}, \mathbf{o}^5_{cmy}) = \operatorname{argmax}_{\mathbf{o}_{cmy}} \{\phi_{avg}(\mathbf{N}^1_{cmy}, \mathbf{o}_{cmy})\}$	maximize ϕ_{avg} over $\mathbf{o}_{cmy}$ given $\mathbf{N}^1_{cmy}$
6	$(\mathbf{N}^2_{cmy}, \mathbf{o}^6_{cmy}) = \operatorname{argmax}_{\mathbf{o}_{cmy}} \{\phi_{max}(\mathbf{N}^2_{cmy}, \mathbf{o}_{cmy})\}$	maximize ϕ_{max} over $\mathbf{o}_{cmy}$ given $\mathbf{N}^2_{cmy}$
7	$(\mathbf{N}^7_{cmy}, \mathbf{o}^7_{cmy}) = \arg \min_{\mathbf{N}_{cmy}, \mathbf{o}_{cmy}} \max \{\phi_{avg}(\mathbf{N}_{cmy}, \mathbf{o}_{cmy})\}$	maximize ϕ_{avg} over $\mathbf{o}_{cmy}$ then minimize over $\mathbf{N}_{cmy}$
8	$(\mathbf{N}^8_{cmy}, \mathbf{o}^8_{cmy}) = \arg \min_{\mathbf{N}_{cmy}, \mathbf{o}_{cmy}} \max \{\phi_{max}(\mathbf{N}_{cmy}, \mathbf{o}_{cmy})\}$	maximize ϕ_{max} over $\mathbf{o}_{cmy}$ then minimize over $\mathbf{N}_{cmy}$

As mentioned before, we need three periodicity matrices and three offset vectors, given in Equations 7.39 and 7.40, for the cyan, magenta, and yellow halftone screens. This implies that there are 18 variables to optimize. Our optimization is based on an exhaustive search strategy with the following restrictions:

- We choose the same fixed density $\mathcal{M}$ for each of the three colorants and consider only those screens for which screen angles are between 20 and 160°, and vector lengths in the periodicity matrices are within the following limits: $1 \leq |\mathbf{n}_1|, |\mathbf{n}_2| \leq \mathcal{M}$. We expect a reasonable solution within these constraints.

Furthermore, we do the following:

- Search over three variables of the periodicity matrix and compute the fourth to satisfy the chosen density $\mathcal{M}$; e.g., if $\mathcal{M}$ is 19 and c_{11}, c_{12}, and c_{21} are 5, 2, and –2, respectively, then $5c_{22} + 4 = 19$, i.e., $c_{22} = 3$.
- Vary the elements of the periodicity matrices from $-\sqrt{\mathcal{M}} + 1$ to $\sqrt{\mathcal{M}} - 1$, and the elements of the offset vectors from 0 to $\sqrt{\mathcal{M}} - 1$.
- Take the offset vector of the cyan colorant to be the null vector. This reduces the number of variables to 13.

7.6.5 *Experimental results*

Our target platform is the HP DeskJet 970C printer. We printed the eight Neugebauer primaries on it and measured their spectral reflectances as discussed in Section 7.2. These primaries are used in the color device model described in Section 7.6.3.1. As discussed in Section 7.2, we use a weighting factor $\kappa = 4$ for our experimental results. For comparison with screens rotated by the conventional (rational tangent) approach, we rotate cyan, magenta, and yellow screens to 0, 33.69, and 68.19°, respectively. The resulting periodicity matrices, offset vectors, and values of the error metrics for the eight cases in Table 7.3 and the conventional screen, for $\mathcal{M} = 16$, are listed in Table 7.4.

Table 7.4 Periodicity Matrices and Offset Vectors for the Screens Described in Table 7.3 for the Case where $\mathcal{M} = 16$

Case	$[\mathbf{N}_c \vert \mathbf{N}_m \vert \mathbf{N}_y]$	$[\mathbf{o}_c \vert \mathbf{o}_m \vert \mathbf{o}_y]$	ϕ_{avg}	ϕ_{max}
1	$\left[\begin{array}{cc\|cc\|cc} 3 & 2 & 3 & 2 & 3 & 2 \\ -2 & 4 & -2 & 4 & -2 & 4 \end{array}\right]$	$\left[\begin{array}{c\|c\|c} 0 & 3 & 1 \\ 0 & 1 & 0 \end{array}\right]$	0.0032	0.0057
2	$\left[\begin{array}{cc\|cc\|cc} 2 & 0 & 2 & 0 & 2 & 0 \\ 3 & 8 & 3 & 8 & 3 & 8 \end{array}\right]$	$\left[\begin{array}{c\|c\|c} 0 & 3 & 1 \\ 0 & 1 & 0 \end{array}\right]$	0.0034	0.0053
3, 4	$\left[\begin{array}{cc\|cc\|cc} 3 & 1 & 2 & 2 & 3 & 1 \\ 2 & 6 & -3 & 5 & 2 & 6 \end{array}\right]$	$\left[\begin{array}{c\|c\|c} 0 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right]$	0.0439	0.0825
5	$\left[\begin{array}{cc\|cc\|cc} 3 & 2 & 3 & 2 & 3 & 2 \\ -2 & 4 & -2 & 4 & -2 & 4 \end{array}\right]$	$\left[\begin{array}{c\|c\|c} 0 & 3 & 3 \\ 0 & 3 & 3 \end{array}\right]$	0.0163	0.0279
6	$\left[\begin{array}{cc\|cc\|cc} 2 & 0 & 2 & 0 & 2 & 0 \\ 3 & 8 & 3 & 8 & 3 & 8 \end{array}\right]$	$\left[\begin{array}{c\|c\|c} 0 & 3 & 3 \\ 0 & 3 & 3 \end{array}\right]$	0.0140	0.0250
7, 8	$\left[\begin{array}{cc\|cc\|cc} 3 & 1 & 3 & 1 & 3 & 2 \\ 2 & 6 & 2 & 6 & -2 & 4 \end{array}\right]$	$\left[\begin{array}{c\|c\|c} 0 & 2 & 0 \\ 0 & 1 & 1 \end{array}\right]$	0.0143	0.0259
Conventional	$\left[\begin{array}{cc\|cc\|cc} 4 & 0 & 2 & -3 & 2 & -2 \\ 0 & 4 & 4 & 2 & 3 & 5 \end{array}\right]$	$\left[\begin{array}{c\|c\|c} 0 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right]$	0.0210	0.0374

The error metrics ϕ_{avg} and ϕ_{max} for the best cases (cases 1 and 2) are comparable. Both worst-case conditions (cases 3 and 4) result in the same periodicity matrices and offset vectors. For the worst case, ϕ_{avg} and ϕ_{max} are about 14 times larger than ϕ_{avg} and ϕ_{max} for the best cases. When the screens designed for the best cases have worst-case registration errors (cases 5 and 6), the error metrics go up by factors between 4 and 5 as compared with the best cases. However, they are still about three times smaller than the error

metrics for the worst case. The optimization for the worst-case registration errors (cases 7 and 8) yields the same set of periodicity matrices and offset vectors. The resulting ϕ_{avg} and ϕ_{max} are four to five times larger than the ones for the best cases and three times smaller than the ones for the worst case overall (cases 3 and 4). For the conventional case, ϕ_{avg} and ϕ_{max} are six to seven times larger than the corresponding errors for the best cases, about two times smaller than that of the worst case, and 30 to 45% larger than the error metrics in cases 5, 6, 7, and 8.

The magnitude of the Fourier transform for the error in $Y_yC_xC_z$ weighted with the appropriate HVS spatial frequency responses for the best (case 1), worst (case 3 or 4), and conventional screens for absorptance = 0.25 are given in Figures 7.26, 7.27, and 7.28, respectively. The frequency units in all three figures are cycles/sample. The spectrum for the best case has a few low-amplitude, high-frequency spectral peaks in the luminance and chrominance channels, which cause no moiré or rosette artifacts. The amplitudes of the spectral peaks in the luminance and chrominance channels go up by a factor of ten for the worst case as compared with the best case. Moreover, these spectral components are in the low-frequency region, which can result in significant moiré artifacts. In the spectra of the error in $Y_yC_xC_z$ for the conventional screen, there are a number of smaller peaks (as compared with the worst case) throughout the low-frequency region. This can result in rosette artifacts.

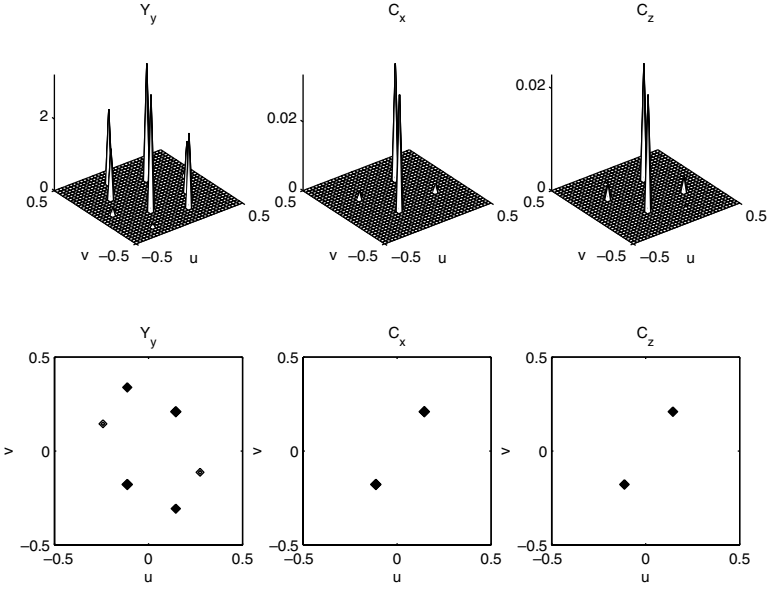


Figure 7.26 Magnitude of the Fourier transform of the error in $Y_yC_xC_z$ weighted with the HVS spatial frequency responses obtained for the screen in case 1 in Table 7.4 for absorptance = 025. The frequency units are cycles per sample.

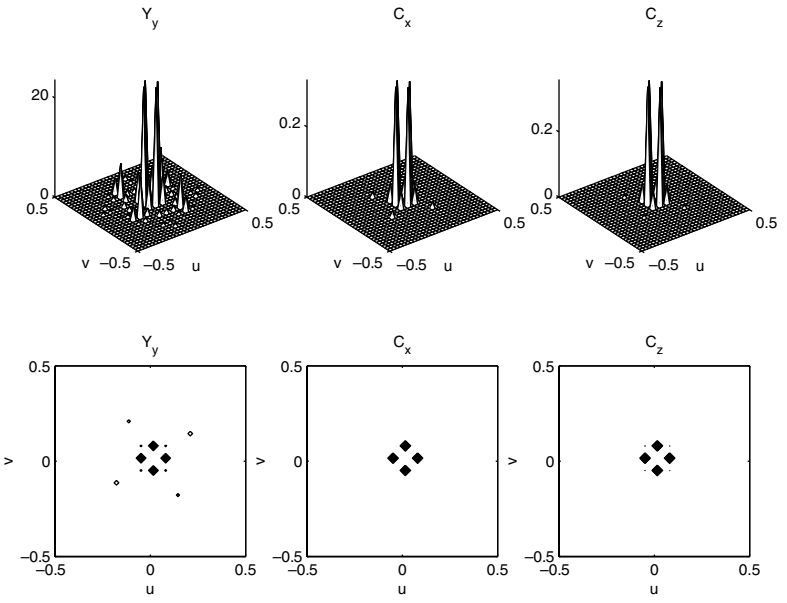


Figure 7.27 Magnitude of the Fourier transform of the error in $Y_y C_x C_z$ weighted with the HVS spatial frequency responses obtained for the screen in case 3 or 4 in Table 7.4 for absorptance = 025. The frequency units are cycles per sample.

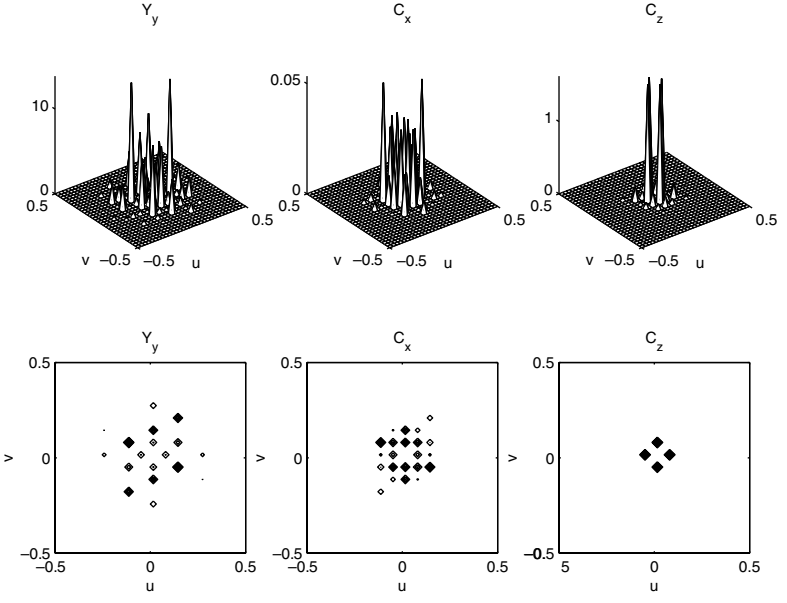


Figure 7.28 Magnitude of the Fourier transform of the error in $Y_y C_x C_z$ weighted with the HVS spatial frequency responses obtained for the conventional screen in Table 7.4 for absorptance = 025. The frequency units are cycles per sample.

To test the performance of the designed screens, we use the following images: (1) gray patches with absorptance ranging from 0.0625 to 0.9375, and (2) the sailboats image. We chose a printer resolution of 150 dpi for printing the images to illustrate the halftone structure. The halftoned images of the gray patches printed with the periodicity matrices and offset vectors in Table 7.4 are shown in Figures 7.29 and 7.30. Magnified scanned textures of the gray patches at an absorptance of 0.25 for the screens in Table 7.4 are also shown in Figure 7.31.

The best screens (cases 1 and 2) give the most smooth regions with no moiré or rosette artifacts. For some levels, the image obtained by the screen designed by case 2 yields slightly better results. The worst-case screen (cases 3 and 4) contains objectionable moiré patterns. Rosette patterns exist in the image halftoned by the screen rotated at conventional angles. There are no moiré or rosette artifacts in screens that are designed for ideal registration but printed with the worst-case registration (cases 5 and 6) and screens that are robust to registration errors (cases 7 and 8). The image quality is also similar. Therefore, it is not important to have good registration of the screens if we use the appropriate periodicity matrix. Of course, it is necessary to have very accurate control of angular alignment.

The halftones of the sailboats image for the periodicity matrix and offset vectors in Table 7.4 are given in Figures 7.32 and 7.33. The worst-case screen introduces objectionable moiré artifacts, while the conventional screen shows rosette formations. The screen obtained for case 2 gives the best performance for this image. The contouring that is visible in these images is due to the low value chosen for the density. If we increase the density of the 3 colorants from 16 to 64, this contouring will be mitigated.

To analyze the effect of the perceptual weighting, we designed screens for $\mathcal{M} = 16$ without the HVS model. The results were dramatically different from the ones with perceptual weighting. The best case yielded a dot-on-dot screen. There was less variance among the metrics. Therefore, the discrimination capability of the algorithm is diminished if HVS models are not used. We also performed designs for $\mathcal{M} = 8$ according to the metrics in Table 7.3. We observed that the relative performance under the various metrics is generally similar to that for $\mathcal{M} = 16$. However, there is a much larger difference between the best and worst cases. It is interesting to note that the periodicity matrices (see Table 7.4) for the different colorants for the best cases (cases 1 and 2) are exactly the same for our chosen density ($\mathcal{M} = 16$). We have observed the same behavior for densities of four and eight. However, it remains to be seen if this effect holds for other values of $\mathcal{M}$ and/or other dot shapes and colorant spectra.

7.7 *Summary and conclusions*

In this chapter, we have addressed the use of color human visual system (HVS) models in the context of color halftoning. Because the use of an HVS model in a color halftoning algorithm will not be effective without also taking

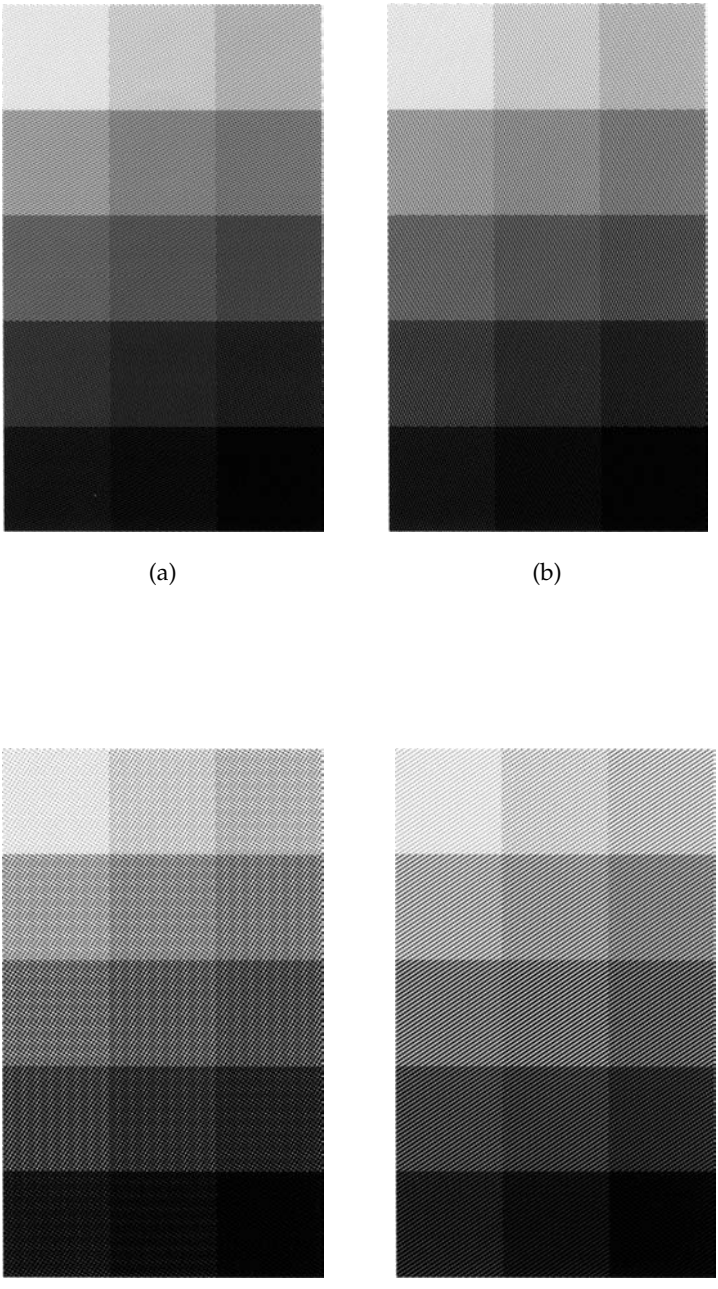
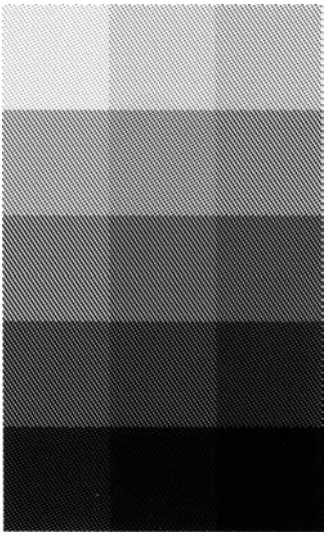
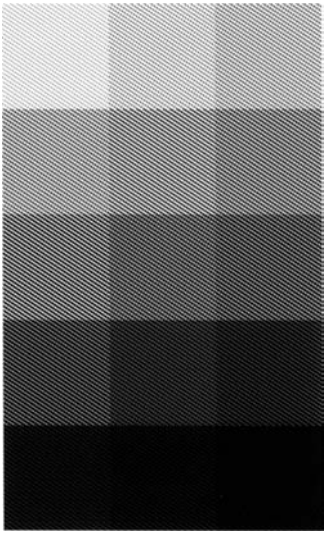


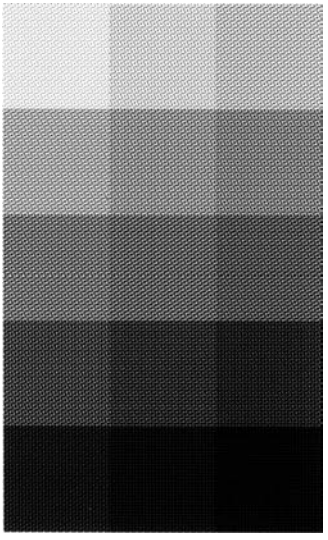
Figure 7.29 ***(See color insert)*** The halftoned images of the gray patches printed with the periodicity matrices and offset vectors in Table 7.4: (a) case 1, (b) case 2, (c) cases 3 and 4, and (d) case 5.



(a)



(b)



(c)

Figure 7.30 ***(See color insert)*** The halftoned images of the gray patches printed with the periodicity matrices and offset vectors in Table 7.4: (a) case 6, (b) cases 7 and 8, and (c) conventional.

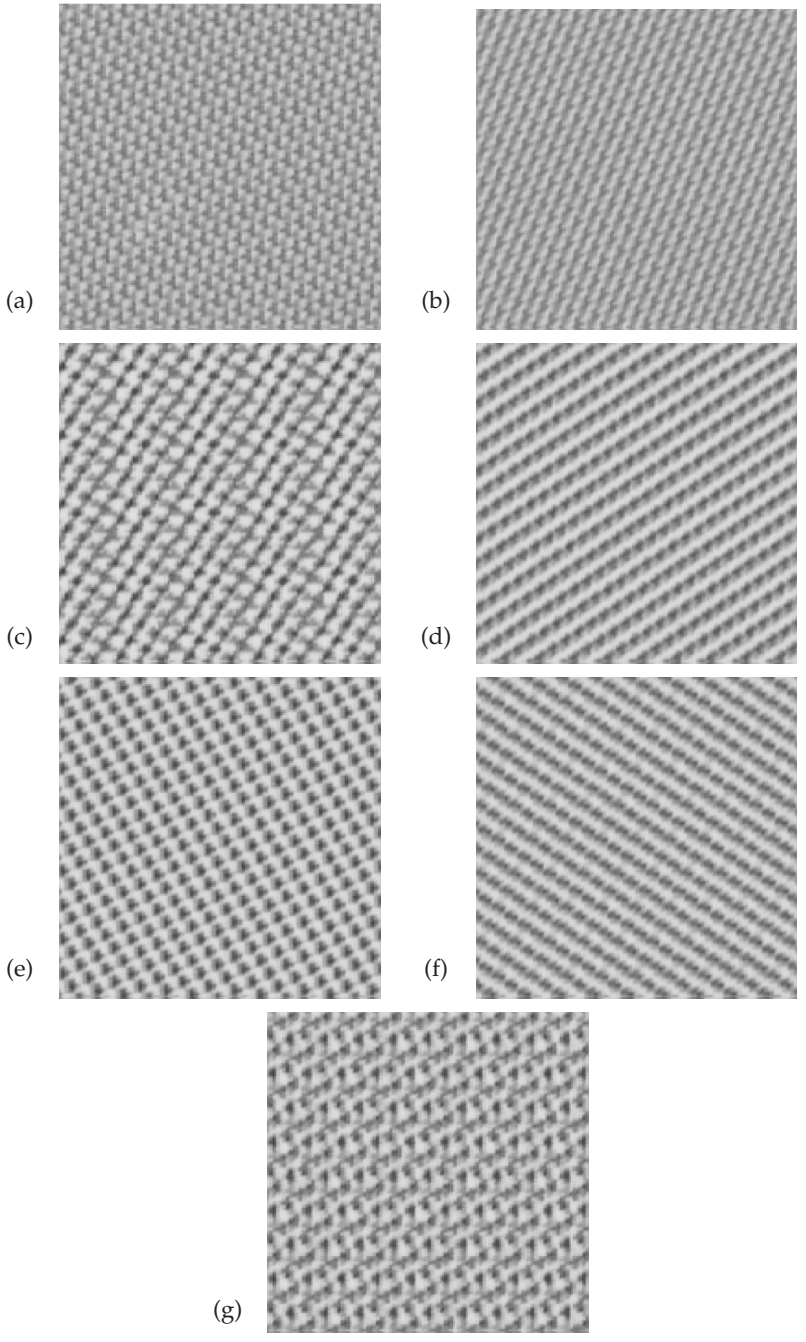
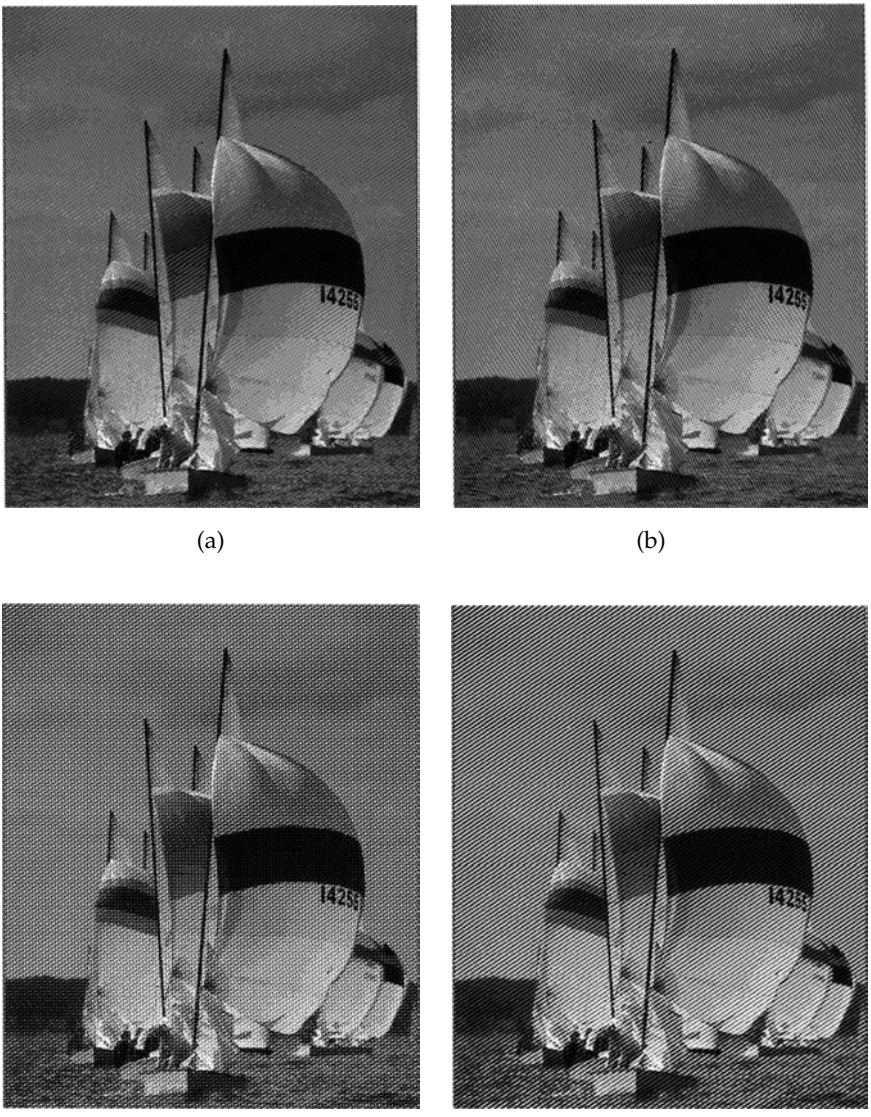


Figure 7.31 ***(See color insert)*** Magnified scanned textures for various screens in Table 7.4 for absorptance = 0.25 (printed at 300 dpi on an HP Deskjet 970Cxi, magnified 6×): (a) case 1, (b) case 2, (c) cases 3 and 4, (d) case 5, (e) case 6, (f) cases 7 and 8, and (g) conventional.

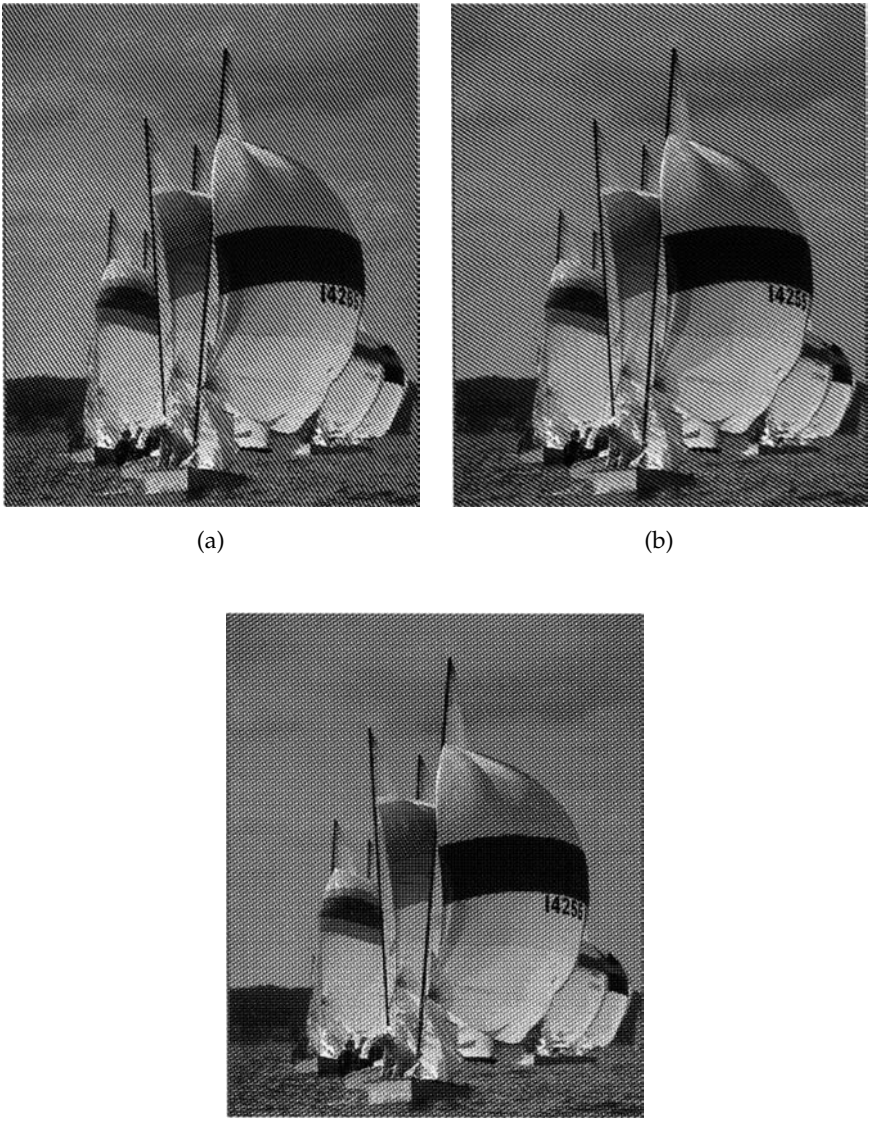


(a) (b)

(c) (d)

Figure 7.32 ***(See color insert)*** Halftones of the sailboats image for the periodicity matrix and offset vectors in Table 7.4: (a) case 1, (b) case 2, (c) cases 3 and 4, (d) case 5.

into account the impact on the final rendered image of the marking engine, colorant, and media, we first reviewed models for color hardcopy systems. We then reviewed spatiochromatic models for the HVS. The model on which we focused consists of three stages. In the first stage, the color image is transformed from a colorant space to a colorimetric space. In the second



(a) (b)

(c)

Figure 7.33 ***(See color insert)*** Halftones of the sailboats image for the periodicity matrix and offset vectors in Table 7.4: (a) case 6, (b) cases 7 and 8, and (c) conventional.

stage, the image is further transformed to an opponent-color space. Finally, within the color opponent space, each channel undergoes spatial filtering, which reflects the broader spatial frequency response of the luminance channel compared with the two chromatic channels. By computing the error between the continuous-tone original image and the halftone rendering of

that image in this space, we obtain the basis for a more preceptually relevant measure of halftone image quality.

We considered the application of this color HVS model to two specific color halftoning tasks: design of an optimal dispersed-dot halftone rendering for a specific continuous-tone image, and design of optimal periodic, clustered-dot screens. Both of these design tasks require a search in a high-dimensional parameter space to find a local minimum of the error metric. Thus, the optimality of our results must be interpreted in the context of both the chosen error metric and the fact that only a local minimum is found. We also briefly reviewed the use of color HVS models in error diffusion.

Our algorithm for dispersed-dot halftoning is based on an extension of the direct binary search (DBS) algorithm to color (CDBS). We find that using the $Y_yC_xC_z$ luminance/chrominance-based space and incorporating a HVS model into the halftoning process does indeed give us visually pleasing color textures. The 2×2 centering dot interaction model improves the color texture rendition, especially in dark and light areas where accurate accounting of the dot interaction of minority colorants with the majority colorant is important.

The DBS algorithm and the efficient search strategies provide faster implementation. Even though CDBS in the $Y_yC_xC_z$ space is a colorimetric halftoning technique that cannot guarantee a desired colorant combination, the calibration algorithm we have developed, cascaded with CDBS, allows us to create color halftones with high texture quality and desired colorant composition. Our calibrated CDBS halftoning algorithm leads to color halftones with very good overall color texture quality and faithful colorant reproduction, except in very dark colors where the underlying HVS model fails to give us a good measure of perceived error.

For design of clustered-dot periodic halftone screens, we have presented a systematic technique based on lattice theory, a color device model, and a spatiochromatic human visual system model for choosing the screen periodicities and offsets that minimize the visibility of moiré and rosette artifacts. We also designed screens that not only suppress moiré and rosette artifacts but are also robust to registration errors. However, at the chosen density, screens that are robust to registration errors do not offer a significant advantage over screens that have been designed for perfect registration, but have worst-case registration errors. Overall, it appears that jointly minimizing the maximum RMSE at any level over the set of periodicity matrices and offset vectors gives the best performance.

A number of extensions can be made to the screen design process. For instance, to get a wider choice of screen periodicities, pulse-width modulation (PWM) can be incorporated. With PWM, each pixel may be in one of several states rather than just two states as is the case for conventional printers. Kacker et al.[107] use PWM in a model for electrophotographic print processes.

Looking into the future, there remains an opportunity to incorporate within color halftoning algorithms more sophisticated HVS models that

include multiple channels and masking effects, such as those proposed by Daly,[108] Lubin,[109] and Taylor et al.[110] These models have been extended to color by Jin et al.[111] and Wu et al.[112] They would couple well with detailed microscopic models for the interaction of light with the media and colorant and would reduce the need for more approximate macroscopic models, such as Neugebauer and Yule–Nielsen, or measurement-intensive printer characterization approaches that are currently used today. The benefits of a more closed-form approach to color halftoning algorithm design would be especially important for reducing the effort required to recalibrate printers and store separate LUTs to account for different combinations of media and colorants.

References

1. Yule, J. A. C., *Principles of Color Reproduction, Applied to Photomechanical Reproduction, Color Photography, and the Ink, Paper, and Other Related Industries,* John Wiley & Sons, New York, 1967.
2. Grum, F. and Bartleson, C. J., Eds., *Optical Radiation Measurements: Color Measurement*, Vol. 2, Academic Press, New York, 1980.
3. Neugebauer, H. E. J., Die theoretischen Grundlagen des Mehrfarbendruckes, *Zietschrift fur Wissenschaftliche Photographie, Photophysik und Photochemie*, 36(4), 73–89, 1937.
4. Yule, J. A. C. and Nielsen, W. J., The penetration of light into paper and its effect on halftone reproduction, *Proc. TAGA*, 4, 66–75, 1951.
5. Viggiano, J. A. S., The color of halftone tints, in *TAGA Proc.*, 647–661, 1985.
6. Heuberger, K. J., Jing, Z. M., and Persiev, S., Color transformations and lookup tables, in *TAGA/ISCC Proc.* 2, 863–881, 1992.
7. Rolleston, R. and Balasubramanian, R., Accuracy of various types of Neugebauer model, in *Proc. First IS&T/SID Color Imaging Conf.*, Scottsdale, AZ, November, 1993, 32–37.
8. Balasubramanian, R., Colorimetric modelling of binary color printers, in *Proc. of IEEE Int. Conf. on Image Proc.*, Washington, D.C., October 23–26, 1995, 327–330.
9. Lee, B. K., Estimation of the Neugebauer model of a halftone printer and its applications, in *IS&T/OSA Optics & Imaging in the Information Age Proceedings*, 1996, 376–379.
10. Chang, S. L., Liu, Y.T., and Yeh, D. Z., A method to estimate fractional areas of Neugebauer primary colors, in *Proc. Fifth IS&T/SID Color Imaging Conf.*, Scottsdale, AZ, November 1997, 97–100.
11. Balasubramanian, R., The use of spectral regression in modeling halftone color printers, in *IS&T/OSA Optics & Imaging in the Information Age Proceedings*, 1996, 372–375.
12. Hua, C. C. and Huang, K. L., Advanced cellular YNSN printer model, in *Proc. Fifth IS&T/SID Color Imaging Conf.*, Scottsdale, AZ, November, 1997, 231–234.
13. Agar, A. U. and Allebach, J. P., An iterative cellular YNSN method for color printer characterization, in *Proc. Sixth IS&T/SID Color Imaging Conf.*, Scottsdale, AZ, November 17–20, 1998, 197–200.

14. Xia, M., Saber, E., Sharma, G., and Tekal, A. M., End-to-end color printer calibration by total least squares regression, *IEEE Trans. on Image Processing*, 8(5), 700–716, 1999.
15. Clapper, F. R. and Yule, J. A. C., Reproduction of color with halftone images, *TAGA Proc*, 1(2), 1955.
16. Wyszecki G. and W. S., Stiles, *Color Science*, John Wiley & Sons, New York, 1982.
17. Kubelka, P., New contributions to the optics of intensely light-scattering materials, Part I, *J. Opt. Soc. of Am.* 38(5), 448–457, 1948.
18. Ruckdeschel, F. R. and Hauser, O. G., Yule–Nielsen effect in printing: a physical analysis, *Appl. Opt.*, 17(21), 3376–3383, 1978.
19. Kruse, B. and Gustavson, S., Rendering of color on scattering media, in *Human Vision and Electronic Imaging*, *Proc. SPIE, 2657,* 1996, 422–431.
20. Arney, J. S., A probability description of the Yule–Nielsen effect I, *J. Imaging Sci. Technol.*, 41(6), 633–636, 1997.
21. Arney, J. S. and Katsube, M., A probability description of the Yule–Nielsen effect II, *J. Imaging Sci. and Technol.*, 41(6), 637–642, 1997.
22. Huntsman, J. R., A new model of dot gain and its applications to a multilayer color proof, *J. Electronic Imaging*, 13(5), 136–145, 1987.
23. Rogers, G. L., Effect of light scatter on halftone color, *J. Optical Soc. Am. A.*, 15(7), 1813–1821, 1998.
24. Emmel, P. and Hersch, R. D., A model for colour prediction of halftoned samples incorporating light scattering and ink spreading, in *Proc. Seventh IS&T/SID Color Imaging Conf.*, Scottsdale, AZ, November 17–20, 1999, 197–200.
25. Yang, L. and Kruse, B., Ink penetration and its effects on printing, in *Proc. SPIE*, 3963, 365–375, 2000.
26. Agar, A. U., A model for halftone color prediction from microstructure, in *Proc. SPIE*, 4300, 416–421, 2001.
27. Roetling, P. G. and Holladay, T. M., Tone reproduction and screen design for pictorial electrographic printing, *J. Appl. Photogr. Eng.*, 5(4), 179–182, 1979.
28. Pappas, T. N., Model-based halftoning of color images, *IEEE Trans. Image Processing*, 6(7), 1014–1024, 1997.
29. Kim, C. Y., Kweon, I. S., and Seo, Y. S., Color and printer models for color halftoning, *J. Electronic Imaging*, 6(2), 166–180, 1997.
30. Lee, C. S., Kim, K. M., Jee, E. J., and Ha, Y. H., Color quantization and dithering method based on HVS characteristics, in *Proc. Third IS&T/SID Color Imaging Conf.*, Scottsdale, AZ, 1995, 90–92.
31. Knuth, D. E., Digital halftones by dot diffusion, *ACM Trans. Graphics*, 6, 245–273, 1987.
32. Lai, J. Z. and Chen, C., Color image halftoning with the dot overlap printer model, in *Proc. 1999 IEEE Intl. Conf. on Image Processing*, Kobe, Japan, 1999, 333–337.
33. Crounse, K. R., Measurement-based printer models with reduced number of parameters, in *Proc. SPIE*, 4663, 121–129, 2002.
34. Wang, S., Knox, K. T., and George, N., Novel centering method for overlapping correction in halftoning, in *ICPS'94: The Physics and Chemistry of Imaging Systems*, 1994, 482–486.
35. Wang, S., Algorithm-independent color calibration for digital halftoning, in *Proc. Fourth IS&T/SID Color Imaging Conf.*, 1996, 75–79.

36. Wang, S., Two-by-two centering printer model with Yule–Nielsen equation, in *IS&T NIP 14: Intl. Conf. on Digital Printing Technologies*, 1998, 302–305.
37. Goertzel G. and Thompson, G. R., Digital halftoning on the IBM 4250 printer, *IBM J. Res. Dev.*, 31(1), 2–15, 1987.
38. Noorlander, C. and Koenderink, J., Spatial and temporal discrimination of ellipsoids in color space, *J. Optical Soc. Am.*, 73, 1533–1543, 1983.
39. Bauml, H. and Wandell, B., The color appearance mixture of gratings, *Vision Res.*, 99(7), 1–100, 1996.
40. Kim, S. H. and Allebach, J. P., Impact of HVS models on model-based halftoning, *IEEE Trans. Image Processing*, 11(3), 258–269, 2002.
41. Mannos, J. L. and Sakrison, D. J., The effects of a visual fidelity criterion on the encoding of image, *IEEE Trans. Information Theory*, 20(4), 526–536, 1974.
42. Daly, S., The visible differences predictor: an algorithm for the assessment of image fidelity, in *Proc. SPIE Human Vision, Visual Processing, and Digital Display III*, 1966, 2–15, 1992.
43. Zhang X. and Wandell, B. A., Spatial extension of cielab for digital color image reproduction, *J. Soc. Info. Display*, 5(1), 61–63, 1997.
44. Mulligan, J. B., Digital halftoning methods for selectively partitioning error into achromatic and chromatic channels, in *Human Vision and Electronic Imaging: Models, Methods, and Applications*, J. P. Allebach and B. E. Rogowitz, Eds., *Proc. SPIE,* 1249, 261–270, 1990.
45. Mulligan, J. B., Vision-based approaches to digital halftoning, in *Proc. IS&T PICS*, April 2002.
46. Zhang, X., Silverstein, D. A., Farrell, J. E., and Wandell, B. A., Color image fidelity metric s-CIELAB and its application on halftone texture visibility, in *IEEE COMPCON97 Symp. Digest*, 1997, 44–48.
47. Agar, A. U. and J. P. Allebach, Model-based color halftoning using direct binary search, in *Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts V, Proc. SPIE,* 3963, 521–535, 2000.
48. Flohr, T. J. , Kolpatzik, B. W., Balasubramanian, R., Carrara, D. A., Bouman, C. A., and J. P. Allebach, Model-based color image quantization, in *Proc. SPIE Human Vision, Visual Processing, and Digital Display IV*, 1913, 270–281, 1993.
49. Lin, Q. and J. P. Allebach, Color FM screen design using DBS algorithm, in *Color Imaging: device-independent Color, Color Hardcopy, and Graphic Arts II, Proc. SPIE,* 3300, 353–361, 1998.
50. Gentile, R. S., Device independent color in postscript, in *Proc. SPIE,* 1913, 419–432, 1993.
51. Mullen, K. T., The contrast sensitivity of human color vision to red–green and blue–yellow chromatic gratings, *Physiol.*, 359, 381–400, 1985.
52. Näsänen, R., Visibility of halftone dot textures, *IEEE Trans. Syst. Man. Cyb.*, 14(6), 920–924, 1984.
53. Sullivan, J. R., Ray, L. A., and Miller, R., Design of minimum visual modulation halftone patterns, *IEEE Trans. Systems Man Cybernetics*, 21(1), 33–38, 1991.
54. Kolpatzik, B. W. and Bouman, C. A., Optimized error diffusion for image display, *J. Electronic Imaging*, 1(3), 277–292, 1992.
55. Balasubramanian, R., Bouman, C. A., and Allebach, J. P., Sequential scalar quantization of vectors: an analysis, *IEEE Trans. on Image Processing*, IP-4, September 1995, 1282–1295.

56. Allebach, J. P., DBS: Retrospective and future directions, in *Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts VI*, R. Eschbach and G. G. Marcu, Eds., *Proc. SPIE,* 4300, 358–376, 2001.
57. Analoui, M. and Allebach, J. P., Model based halftoning using direct binary search, in *Human Vision, Visual Processing, and Digital Display III, Proc. SPIE,* 1666, 96–108, 1992.
58. Mulligan, J. B. and Ahumada Jr., A. J., Principled halftoning based on human visual models, in *Human Vision, Visual Processing, and Digital Display III, Proc. SPIE,* 1666, 109–121, 1992.
59. Pappas, T. N. and Neuhoff, D. L., Least-squares model-based halftoning, in *Human Vision, Visual Processing, and Digital Display III, Proc. SPIE,* 1666, 165–176, 1992.
60. Analoui, M. and Allebach, J. P., Model-based halftoning using direct binary search, in *Proc. of SPIE/IS&T Symp. on Electronic Imaging Science and Tech.*, San Jose, CA, February 1992, 96–108.
61. Lieberman, D. J. and Allebach, J. P., Efficient model based halftoning using direct binary search, in *Proc. 1997 IEEE Int. Conf. on Image Processing*, Santa Barbara, CA, October 26–29, 1997.
62. Daly, S., The visible differences predictor: an algorithm for the assessment of image fidelity, in *Proc. SPIE Human Vision, Visual Processing, and Digital Display III*, 1966, 2–15, 1992.
63. Klassen, R. V. and Bharat, K., Vector diffusion in a distorted colour space, in *Proc. IS&T 47th Annual Conf.*, 1994, 489–491.
64. Pappas, T. N., Dong, C., and Neuhoff, D. L., Measurement of printer parameters for model-based halftoning, *J. Imaging Technol.*, 2(3), 193–204, 1993.
65. Gennetten, K. D., RGB to CMYK conversion using three-dimensional barycentric interpolation, in *Proc. SPIE*, 1909, 116–126, 1993.
66. Shaked, D., Arad, N., Fitzhugh, A., and Sobel, I., Ink relocation for color halftones, in *Proc. IS&T Image Processing, Image Quality, and Image Capture Systems Conf.*, Portland, OR, 1998, 340–343.
67. Sharma, G. and Trussell, H. J., Digital color imaging, *IEEE Trans. Image Processing*, 6(7), 901–932, 1997.
68. Haneishi, H., Suzuki, T., Shimoyama, N., and Miyake, Y., Color digital halftoning taking colorimetric color reproduction into account, *J. Electronic Imaging*, 5(1), 97–106, 1996.
69. Marcu, G., and Abe, S., An error diffusion algorithm for arbitrary set of output color with application to textile sewing, in *Proc. SPIE, Color Hard Copy and Graphic Arts IV*, San Jose, CA, 2413, 375–384, 1995.
70. Shu, J., Error diffusion with vivid color enhancement and noise reduction, in *Proc. SPIE*, San Jose, CA, 2657, 464–470, 1996.
71. Shu, J. and Boyce, J., Adaptive color error diffusion to improve halftone smoothness, in *Proc. SPIE*, San Jose, CA, 3018, 308–311, 1997.
72. Akarun, L., Yardimci, Y., and Cetin, A. E., Adaptive methods for dithering color images, *IEEE Trans. Image Pro.*, IP-6(7), 950–956, 1997.
73. Bozkurt, G., Yardimci, Y., Arikan, O., and Cetin, E., QR-RLS algorithm for error diffusion of color images in *Proc. 1998 IEEE Intl. Conf. on Image Processing*, Chicago, IL, 1998.
74. Lau, D. L., Arce, G. R., and Gallagher, N. C., Digital color halftoning with generalized error diffusion and multichannel green-noise masks, *IEEE Trans. Image Processing*, 9(5), 923–935, 2000.

75. Knox, K. and Eschbach, R., Threshold modulation in error diffusion, *J. Electronic Imaging*, 2(3), 185–192, 1993.
76. Levien, R., Well tempered screening technology, in *Proc. IS&T 3rd Tech. Symp. Prepress, Proofing and Printing*, Chicago, IL, November 1993.
77. Damera-Venkata, N. and Evans, B. L., Design and analysis of vector color error diffusion halftoning systems, *IEEE Trans. Image Processing*, 10(10), 1552–1565, 2001.
78. Poirson, A. B. and Wandell, B. A., Pattern-color separable pathways predict sensitivity to simple colored patterns, *Vis. Res.*, 36, 515–526, 1996.
79. Oster, G., *The Science of Moiré Patterns*, 2nd ed., Edmund Scientific Co., 1969.
80. Allebach, J. P. and Liu, B., Random quasiperiodic halftone process, *J. Opt. Soc. Am.*, 66(9), 909–917, 1976.
81. Allebach, J. P., Random nucleated halftone screen, *Photo. Sci. Eng.*, 22(2), 89–91, 1978.
82. Allebach, J. P. and Liu, B., Analysis of halftone dot profile and aliasing in the discrete binary representation of images, *J. Opt. Soc. Am.*, 67(9), 1147–1154, 1977.
83. Allebach, J. P. and Stradling, R. N., Computer-aided design of dither signals for binary display of images, *Appl. Opt.*, 18(15), 2708–2713, 1979.
84. Roetling, P. G., Halftone method with edge enhancement and moiré suppression, *J. Opt. Soc. Am.*, 66(10), 985–989, 1976.
85. Levien, R., Moiré suppression screening, in *Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts V, Proc. SPIE*, 3963, 402–407, 2000.
86. Rao, T. S. and Arce, G. R., Halftone patterns for arbitrary screen periodicities, *J. Opt. Soc. Am. A*, 5(9), 1502–1511, 1988.
87. Rao, T. S., Arce, G. R., and Allebach, J. P., Analysis of ordered dither for arbitrary sampling lattices and screen periodicities, *IEEE Trans. Acoust., Speech Signal Processing*, 5(9), 1502–1511, 1988.
88. Lau, D. L., Arce, G. R., and Gallagher, N. C., Green noise digital halftoning, *Proc. IEEE*, 86(12), 2424–2442, 1998.
89. Amidror, I., The moiré phenomenon in colour separation, in *Proc. 2nd Int. Conf. Raster Imaging and Digital Typography*, R. A. Morris and J. André, Eds., Cambridge University Press, New York, 1991, 98–119.
90. Oster, G., Wasserman, M., and Zwerling, C., Theoretical interpretation of moiré patterns, *J. Opt. Soc. Am.*, 54(1), 169–175, 1964.
91. Amidror, I., Hersch, R. D., and Ostromoukhov, V., Spectral analysis and minimization of moire patterns in color separation, *J. Elect. Imag.*, 3(3), 295–317, 1994.
92. Kaji, M., Satou, Y., and Tajima, J., A construction method of digital screen sets: Realization of moiré-free rational tangent screens by using the multi-unit area design method, in *Proc. IS&T PICS*, 1998, 349–357.
93. Rodriguez, M., Graphic arts perspective on digital halftoning, in *Human Vision, Visual Processing, and Digital Display V, Proc. SPIE*, 2179, 144–149, 1994.
94. Schoppmeyer, J., Screen Systems for Multicolor Printing, U.S. Patent No. 4,537,470, 1985.
95. Wang, S., Fan, Z., and Wen, Z., Nonorthogonal halftone screens, in *Proc. IS&T NIP 18: Int. Conf. on Digital Printing Technologies*, San Diego, CA, October 2002.
96. Rahgozar, M. A., and Allebach, J. P., A general theory of time sequential sampling, *Signal Process.*, 28(3), 253–270, 1992.

97. Willis, N. P. and Bresler, Y., Lattice theoretic analysis of time-sequential sampling of spatiotemporal signals I, *IEEE Trans. Inform. Theory*, 43(1), 190–207, 1997.
98. Willis, N. P. and Bresler, Y., Lattice theoretic analysis of time-sequential sampling of spatiotemporal signals II: Large space-bandwidth product asymptotics, *IEEE Trans. Inform. Theory*, 43(1), 208–220, 1997.
99. Viscito, E. and Allebach, J. P., The analysis and design of multidimensional FIR perfect reconstruction filter banks for arbitrary sampling lattices, *IEEE Trans. Circuits and Syst.*, 38(1), 29–41, 1991.
100. Dudgeon, D. and R. M. Mersereau, *Multidimensional Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1984.
101. Dubois, E., The sampling and reconstruction of time-varying imagery with application in video systems, *Proc. IEEE*, 73(4), 502–522, 1985.
102. Artin, M., *Algebra*, Prentice-Hall, Englewood Cliffs, NJ, 1991.
103. O'Rourke, J., *Computational Geometry in C*, 2nd ed., Cambridge University Press, New York, 1998.
104. Wesner, J. W., Screen patterns used in reproduction of continuous-tone graphics, *Appl. Opt.*, 13, 1703–1710, 1974.
105. Holladay, T. M., An optimum algorithm for halftone generation for displays and hard copies, in *Proc. SID*, 1980, 185–192.
106. CIE, *Colorimetry*, CIE Pub. No. 15.2, Center Bureau CIE, Vienna, Austria, 1986.
107. Kacker, D., Camis, T., and Allebach, J. P., Electrophotographic process embedded in direct binary search, *IEEE Trans. Image Processing*, 11, 243–257, 2002.
108. Daly, S., The visible differences predictor: an algorithm for the assessment of image fidelity, in *Digital Images and Human Vision*, A. B. Watson, Ed., MIT Press, Cambridge, MA, 1993, 179–205.
109. Lubin, J., A visual discrimination model for imaging systems design and evaluation, in *Vision Models for Target Detection and Recognition*, E. Peli, Ed., World Scientific, Singapore, 1995, 245–283.
110. Taylor, C. C., Pizlo, Z., Allebach, J. P., and Bouman, C. A., Image quality assessment with a Gabor pyramid model of the human visual system, in *Human Vision and Electronic Imaging, Proc. SPIE,* 3016, 58–69, 1997.
111. Jin, E. W., Feng, X. -F., and Newell, J., The development of a color visual difference model (CVDM), in *Proc. IS&T PICS*, May 1998, 154–158.
112. Wu, W., Pizlo, Z., and Allebach, J. P., Color image fidelity assessor, in *Proc. IS&T Image Processing, Image Quality, Image Capture, and Systems Conf.,* April 2001.

chapter eight

Compression of color images

Ricardo de Queiroz
Xerox Corporation

Contents

In this chapter, we intend to cover the basic aspects of color image compression. Basic aspects of color images can be found elsewhere in this book, and compression details can also be easily found in the literature. Even though we give a very brief introduction to compression methods, we intend to explore the issues pertaining to the intersection of the two topics (compression and color) without exploring them individually.

8.1 Compression basics

Image compression methods rely on the removal of information within images to reduce the amount of data necessary to represent them.[1–7] The information to be removed is usually characterized as one of two classes,

statistically redundant or *visually irrelevant.* The removal of (statistically) redundant data often yields reversible or lossless compression; i.e., the image data can be completely retrieved from the compressed data. On the other hand, removing the (visually) irrelevant information inflicts losses on the reconstructed image; it is hoped, however, that such losses are not objectionable for a given destination viewer.

Statistical redundancy is the extra amount of bits used to represent a given sequence of symbols. For example, if a sequence of symbols from the alphabet {A,B,C} is to be encoded, there are several ways to encode the symbols into binary code words. Examples of code words are {00, 01, 10}, {001, 100, 111}, {0, 10, 11}, etc. Some representations are clearly longer than others. In other words, if we start from a given representation, by changing the binary encoding, one might represent the same information with fewer bits. For example, if the probabilities of occurrence are

$p(\mathrm{A}) = 0.5$
$p(\mathrm{B}) = 0.2$
$p(\mathrm{C}) = 0.3$

then the average bit rate R per symbol achievable by each code is

$$\left.\begin{array}{l} A = 00 \\ B = 10 \\ C = 01 \end{array}\right\} R = 2 \text{ bits/symbol} \qquad \left.\begin{array}{l} A = 0 \\ B = 10 \\ C = 11 \end{array}\right\} R = 1.5 \text{ bits/symbol}$$

which clearly shows how the rate could be reduced by 25% by only modifying the code. There are several means by which one can encode a particular sequence of symbols. The above examples are for instantaneous codes where each symbol is assigned a unique code. Instantaneous codes are those for which no code word is a prefix of another code word. (For example, 01 is a prefix of 010.) Also, the above examples are block codes; i.e., each symbol is independently mapped to a code string and vice versa.[8]

If the symbols have probabilities p_i, the entropy of the source is defined to be

$$H = -\sum_i p_i \log p_i \tag{8.1}$$

One cannot encode a source with average rate below the entropy rate without losses; i.e., H is the lower bound for coding a particular source.[8] Furthermore, one can always construct codes with length L_i such that the average length $\bar{L} = \Sigma_i L_i p_i$ is bounded by $H \le \bar{L} \le H + 1$.

Huffman codes are typical examples of uniquely decodable and instantaneous block codes that are optimized for a particular source.[2,3,7,8] However,

as any block code, Huffman codes cannot have less than one bit per symbol. So, when the entropy is far below one bit per symbol, Huffman codes are not so efficient. In fact, one can encode the symbols at an average rate very close to H, even if $H << 1$. One example is the arithmetic coder,[3] which combines multiple symbols and codes to achieve higher performance.

Insofar as there is no distortion to the data, this reduction in the data size can be viewed as statistical redundancy extraction. Yet another type of information that can be removed is the visually irrelevant one. To remove it, the image data are manipulated (preprocessed) so as to improve the statistical redundancy extraction step. However, an attempt is made to make all image changes invisible to the final observer. If the changes are perceptible, they are supposed to have low visibility, at least. The most common method to remove visual redundancy is to remove image details of little importance or frequency components that are not discernible. The removal of information from the image data is performed through the quantization operation.

A quantizer $\boldsymbol{Q}$ maps the input (a random variable, e.g., X) into another symbol of a reduced set. For example, if X assumes numbers in the real line, it can be broken into q intervals by defining $q - 1$ decision levels, t_n, in the real axis that effectively divide the axis into q intervals. In this example, X is assigned to level k if $\{X = x \mid t_{k-1} \leq x < t_k\}$. The inverse quantization operation ($\boldsymbol{Q}^{-1}$) restores a number in the domain of X from the received coded level number k. Typically, the reconstructed value lies within the interval to where the original value belongs; i.e., the reconstructed value for the kth level $\widehat{x}$ would lie within the interval $[t_{k-1}, t_k)$. In other words, quantization maps the input (continuous or discrete) to a set of integer numbers, while the inverse quantization maps the range back into the input domain, i.e.,

$$X \xrightarrow{Q} \ell \xrightarrow{Q^{-1}} \widehat{X}$$

As a many-to-one mapping, quantization implies losses. However, losses are the price paid to largely reduce the number of possible input values and the data entropy. A typical quantization is the uniform one with center reconstruction where

$$x_q = \text{round}\left(\frac{x}{\Delta}\right) \qquad \widehat{x} = x_q + 1/2\Delta \tag{8.2}$$

This form of quantization can be very efficiently implemented and is often used in compression standards because of its low complexity. A variation of the uniform quantizer is the one including a small *dead zone* in the middle,

$$x_q = \left\lfloor \frac{x}{\Delta} \right\rfloor \qquad \widehat{x} = (x_q + 1/2)\Delta \tag{8.3}$$

where $\lfloor \cdot \rfloor$ means a round-off toward zero or "floor" function. The function of the dead zone is to nullify numbers close to zero, which is an efficient means to remove noise.

To remove only statistical redundancy implies lossless compression; i.e., the data after decompression are bit by bit identical to the original. When lossy compression is applied, the data cannot be recovered thereafter, thus implying the use of some form of data quantization. Lossy compression is much more efficient than lossless compression; i.e., by accepting data losses (not too visible), much more compression can be applied to a particular image. This is a general trade-off in compression: rate vs. distortion. Example coders will be discussed later on as well as rate-distortion paradigms.

As far as compression techniques are concerned, this chapter is merely introductory, and the reader is referred to the many texts in image compression to understand compression techniques in more detail.[1–7] The central focus of this chapter is to cover representation of color image information and its interaction with the compression.

8.2 Compression models

8.2.1 Transform coding

Transform coding is perhaps the most popular method of compression. A typical compression system based on transform coding is depicted in Figure 8.1. The input image data is transformed, quantized, and encoded. The encoding step provides lossless compression and simply converts numbers (symbols) to some binary representation in a reversible manner. The quantization step implies losses but does not compress the data. In fact, it controls the compression and the distortion. The transform step does not control distortion or compression, nor does it provide the compression by encoding the data. However, it is the key enabler for successful image compression. Its job is to convert the image pixels into something that would make sense to quantize and encode. Figure 8.1b depicts the transform and quantization steps in greater detail. The data are transformed, and each transformed sample may undergo different quantization. The process of selecting quantizers is historically tied to the process of *bit allocation*.[2,4,6,7] This is so because, in the past, each quantizer used to be associated with its own *encoder;* i.e., a quantizer result was directly mapped into output binary codes. In this situation, the transform output was directly related to the compressed bit rate. In fact, under some reasonable conditions, the compression efficiency can be related to *energy compaction*,[7,9] which is a measure of how much energy is concentrated in a few transformed coefficients. A useful measure can be the ratio of geometric by arithmetic means of the energy of each transformed sample if we apply orthogonal transforms.[7] In this scenario, for very simple quantization and coding, with a few additional assumptions,[7] the optimal transform can be found as the one that provides the most

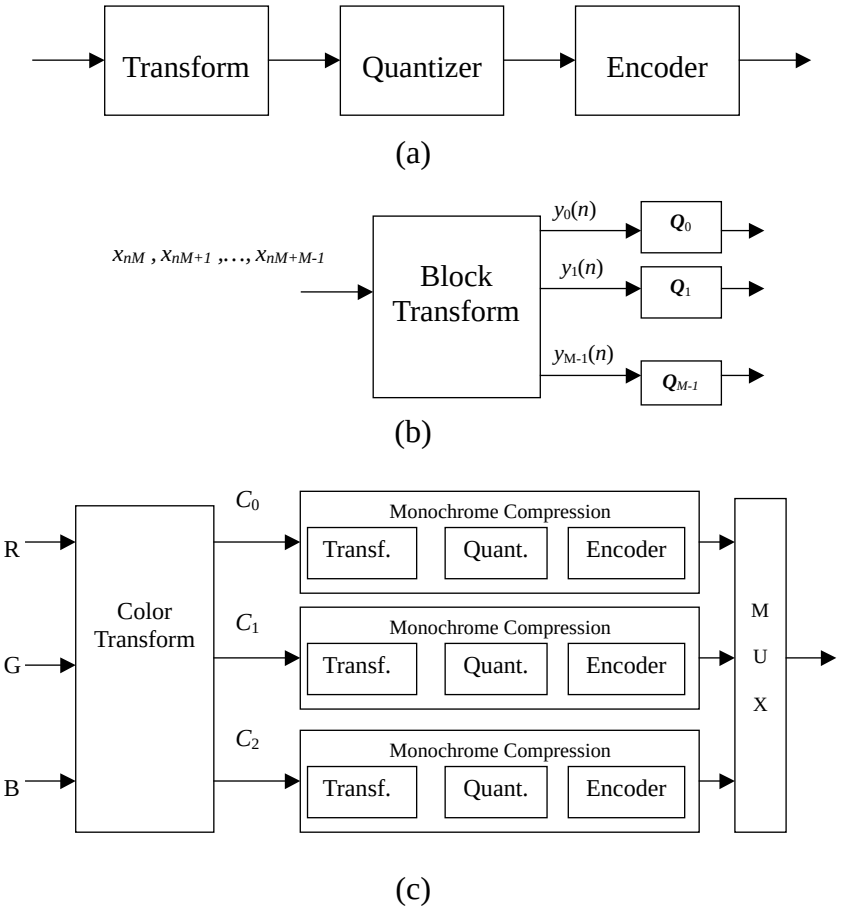


Figure 8.1 Basic steps of typical transform coding systems: (a) and (b) Details of the transform and quantization steps. The transformed samples for each channel commonly are processed using different quantizers. (c) Typical coding system for color images, exemplified for an RGB input image. After a pixel-wise color transform, each resulting color plane is compressed independently, while the resulting compressed bit streams are multiplexed to produce the output.

energy compaction. The Karhunen–Loeve transform (KLT)[2,4,7,9] achieves optimal compaction. If, in the past, each quantizer was associated with its encoding method, it is a modern practice to analyze multiple quantized samples at once before entropy coding the whole quantized data stream. Even so, the connection between energy compaction and compression efficiency is clear from practice.

As an example of how energy compaction can be useful, let us consider the example in Figure 8.2. Recall that an orthogonal transformation is a simple rotation of the input space. Refer to Figure 8.2, where two contiguous samples in an image are being computed. For every pair of pixels, one point

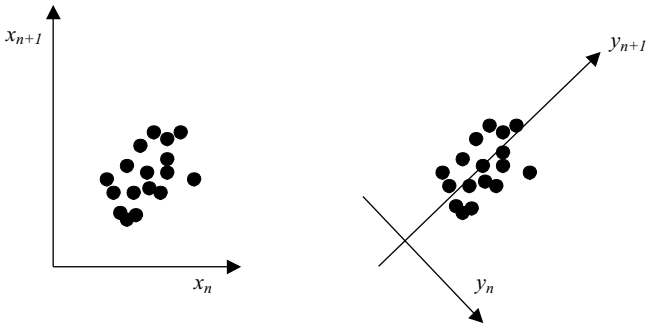


Figure 8.2 An orthogonal transformation is a simple axis rotation. In an image, neighbor pixels (x_n, x_{n+1}) are likely to have similar values. A rotation of $\pi/4$ provides a representation that is better aligned with the data; i.e., there is compaction of energy in one of the variables (y_{n+1}).

in plotted in the (x_n, x_{n+1}) plane. Because neighbor pixels are likely to be similar, it is expected that points would cluster along a diagonal. Hence, an axis rotation of $\pi/4$ might align one of the resulting axes to the data cluster. With this, one achieves compaction of the energy necessary to represent the data around one axis (y_{n+1}) so that y_n tends to have small values. If one had the ability to keep only one of the samples, it is clear that, in the (y_n,y_{n+1}) domain, the distortion incurred by discarding (setting to 0) y_n is much smaller than discarding either x_n or x_{n+1}. Hence, the advantage of providing a transform step that yields energy compaction becomes clear.

The $\pi/4$ rotation we just described is accomplished, for example, with a size-two discrete cosine transform (DCT).[9] In practical compression, vectors with eight samples are transformed using the size-eight DCT.[9] Let **X** be a matrix containing a block with $M \times M$ pixels. In two dimensions, pixel blocks are transformed by a separable transform such that

$$\mathbf{Y} = \mathbf{D}\mathbf{X}\mathbf{D}^T \qquad \mathbf{X} = \mathbf{D}^T\mathbf{Y}\mathbf{D} \tag{8.4}$$

where **Y** is the matrix with the transformed data and **D** is the size-M DCT matrix whose entries $\{d_{ij}\}$ are given by

$$d_{ij} = \sqrt{\frac{2}{M}}\alpha_i \cos\left(\frac{(2j+1)i\pi}{2M}\right)$$

$$\alpha_0 = \frac{1}{\sqrt{2}} \text{ and } \alpha_i = 1 \quad \text{for} \quad i > 0 \tag{8.5}$$

The DCT is asymptotically optimal for some image models and performs well for most images.[9] In fact, for an order-1 autoregressive model, the DCT

approaches the KLT as the correlation between samples increases. The DCT works well to compact the input energy into few transformed samples, as illustrated in Figure 8.3.

Another powerful transform tool in image compression is the wavelet transform.[10,11] The wavelet transform is based on an elementary two-channel filter bank. Each bank decomposes the image into lowpass and highpass sub-bands. The filter bank has a lowpass and a highpass filter, and each is followed by a 2:1 decimator. At every stage of the two-channel filter bank, the number of input samples is maintained; i.e., for N input samples, there are $N/2$ lowpass output samples and $N/2$ highpass ones. The decomposition is also said to be the analysis. To recompose the original signal from the lowpass and highpass sub-bands, in a process called the *synthesis*, one uses

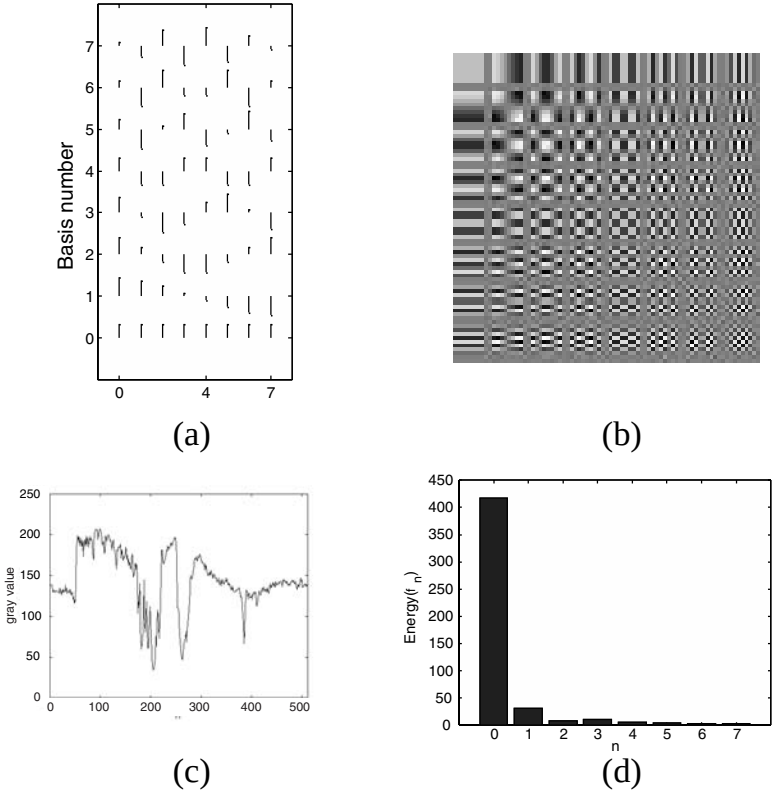


Figure 8.3 The eight-basis functions of the DCT in the one-dimensional case (a) and its two-dimensional separable counterpart (b). The pixels of the 8×8 block are projected into each of the bases shown. The projection enables energy compaction. For example, the signal shown in (c) is a scan line of a test image. By breaking the vector into eight sample segments and transforming each one, the energy of the DCT coefficients is depicted in (d). Note the concentration of energy in a few low-frequency coefficients.

1:2 up-samplers followed by the lowpass and highpass filters. The filtered output for both channels is added to resynthesize the input data. For the two-dimensional case, the image is decomposed into four decimated sub-bands by applying analysis filter banks in each of horizontal and vertical directions as depicted in Figure 8.4a for the analysis and Figure 8.4b for the

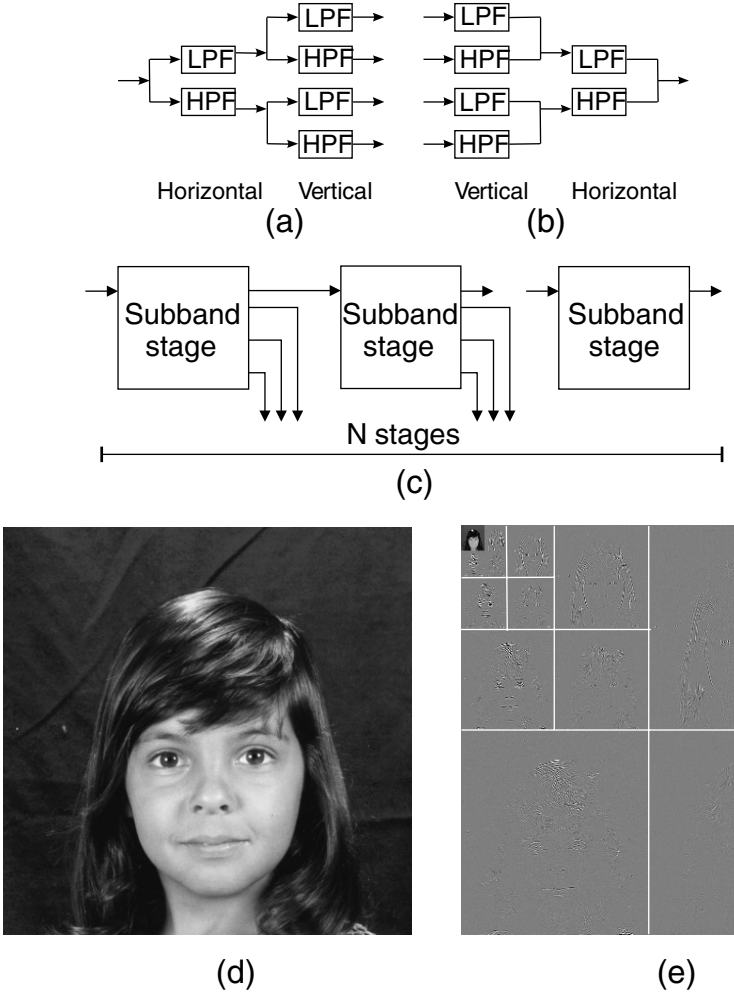


Figure 8.4 Wavelet transform of an image is based on the elementary two-channel filter bank, which, in two dimensions, decomposes the image into lowpass and highpass sub-bands in each of the horizontal and vertical directions (a). The decomposition (analysis) is achieved through filters, followed by 2:1 down-samplers. The synthesis is formed by 1:2 up-samplers, followed by filters. The wavelet transform is the association of filter banks (c) such that the lowpass of one stage is input into another stage. In this way, the image in (d) is decomposed into the sub-bands depicted in (e).

synthesis section. The wavelet transform is an association of filter banks as in Figure 8.4c such that the lowpass output of one stage is input to another stage. In this way, the image in Figure 8.4d is decomposed into the sub-bands depicted in Figure 8.4e.

The different sub-bands contain information at a different orientation and different scale. However, unlike the DCT, higher-frequency bands are wider and have more samples. In other words, some bands have poor frequency resolution (wide bandwidth) but high spatial resolution (better feature location). Some other bands have better frequency resolution but poor spatial location (narrow bandwidth and just a few samples to represent the whole image). This resolution trade-off is an attractive feature of the wavelet transform and has been extensively studied.[10,11]

In a manner similar to block transforms, the wavelet transform aims at transforming the image data into a representation where a few coefficients convey most of the information to reconstruct the image. Referring back to Figure 8.1, the transform is used to convert the image data to a more suitable representation. It is left to the quantization step to perform the job of actually reducing the amount of information to be conveyed to the decoder. Then, the encoder is left with the job of actually *compressing* the quantized transformed data. For example, for 8-bit image data, the transformed samples can be floating point numbers, which are quantized to integer numbers using a fixed-point representation of, for example, 16 bits (a 1:2 expansion so far). The entropy encoding step is the one that achieves the compression, properly enabled by the previous steps.

8.2.2 Predictive coding

In predictive coding,[2,6] the image data are not compressed in the manner described in Figure 8.1. Instead, one predicts the image pixels and computes a prediction error. The prediction error is actually the information that is quantized and encoded. In general, predictive coding is similar to DPCM systems,[2] which were commonly used before computer technology made transform coding affordable. Figure 8.5 depicts a typical predictive coding system. The actual pixel is predicted somehow from information conveyed through the past (already processed) pixels, and only the prediction error is quantized and encoded. The decoder applies the same prediction step as the encoder. Given the predicted value, the decoder simply integrates the error to the predicted value so as to produce the reconstructed pixel. A feedback loop is required at the encoder side to synchronize encoder and decoder. However, predictive systems are often used for "lossless" compression; i.e., the data are not quantized and, hence, there is no need for the feedback loop. There are several methods for adaptive or nonadaptive prediction. Typically, the pixel illustrated in Figure 8.5c and labeled "X" is predicted from its neighbors (pixels labeled "A" through "H") using some weighted combination. Commonly, the immediate neighbors (pixels "A" through "D") are used to predict the actual pixel "X."

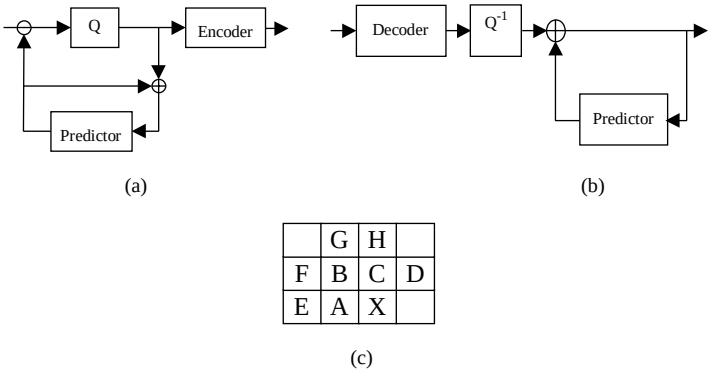


Figure 8.5 In predictive coding (DPCM), the pixels are predicted from past values, and the prediction error is quantized, encoded, and transmitted (a). The receiver (b) decodes the error and adds it to its own predicted value. For synchronizing transmitter (a) and receiver (b) predictions, the transmitter incorporates a local decoder in the form of a feedback loop. A prediction template is illustrated in (c). In this template, pixels "A" through "H" are used to predict pixel "X."

8.2.3 Rate-distortion trade-off

To select compression systems or features, one needs to somehow evaluate the compression performance. If we ignore computational complexity, we can look at "how much compression can we achieve" and at "what is the image quality" as the major issues in compression. The *cost* of compressing an image can be expressed in the amount of data needed to record the information, i.e., the amount of bits consumed. In compression jargon, this is the *rate (R)* achieved by compressing a given image. The *benefit* of compressing the image is the image quality, or how well we approximate the original image, assuming possible lossy compression. By defining some measure of distortion D as the distance between the images before and after compression, we can quantify the compression process as a rate-distortion trade-off.[12] Commonly, the more distortion is allowed, the lower the achievable rate. Conversely, the more bits are spent (higher R), typically, the more accurate the representation (lower D).

Compressors are guided by parameters that are set by the user or application. Examples of parameters are quantizer steps, etc. For a given image, for each parameter choice, the (de)compression system achieves a particular rate R_0 and a particular distortion D_0; i.e., a instantiation of the coder is a point on the RD plane. Let the set of all realizable RD points be Ω. If Ψ is the set of all parameters to be adjusted, the coder can be seen as a map from the domain of Ψ to points in Ω. We always want to minimize rate and distortion at the same time; that is, we want to have better quality at higher compression ratios:

$$\min_{\Psi} D|_{R \le R_T} \quad \text{or} \quad \min_{\Psi} R|_{D \le D_T}, \quad R, D \in \Omega \tag{8.6}$$

The points satisfying the above equation make the lower (typically) convex hull (LCH) of Ω as illustrated in Figure 8.6. Theoretically, coders should operate at an operational point on the LCH for best performance. Typically, if optimization is involved, to get Equation 8.6, one can minimize the cost function $R + \lambda D$. If we move a line with inclination $-1/\lambda$, the first point in Ω it touches is the point that minimizes $R + \lambda D$. This point belongs to the LCH. Actually, varying λ would move the operational point on the LCH.

The important message is that compression performance has a two-dimensional range. It does not matter if a coding method compresses more than another without regard to the distortion. Even if both rate and distortion changes are computed, a fair comparison between methods would require comparing *RD* curves and regions. One good method is to fix either R or D and compare the other dimension.

8.2.4 Distortion measure

The computation of distortion involves some measure of the distance between the original and decompressed images. A common measure of distortion between images is the mean squared error (MSE). If the N color channels of the original image are denoted as $C_k(m, n)$ for a pixel at position (m, n), then

$$MSE = \frac{1}{N_p N} \sum_{k} \sum_{m,n} \left(C_k(m,n) - \hat{C}_k(m,n)\right)^2 \tag{8.7}$$

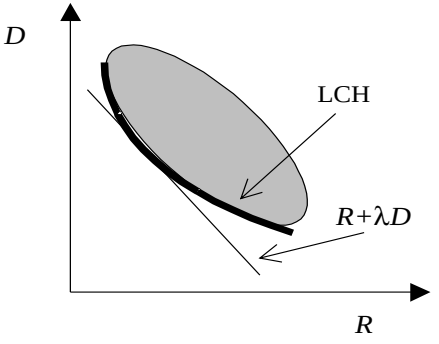


Figure 8.6 In all lossy coders, there is an RD trade-off. While Ω is the space of all attainable RD points, there is typically a lower convex hull (LCH) to the distribution. The LCH contains the desirable operational points. Often, the cost function is chosen as $R + \lambda D$, which is minimized by the point in Ω that first touches the line with inclination $-1/\lambda$.

where $\hat{C}_k$ denotes the reconstructed image planes, and there are N_P pixels in the image. In practice, the MSE is often presented in another form: the peak signal-to-noise ratio (PSNR). PSNR is by far the most commonly used distortion measure for images and is defined for eight bits per pixel (bpp) images as

$$PSNR = 10 \log_{10}\left(\frac{255^2}{MSE}\right) \tag{8.8}$$

While PSNR is easy to compute, it provides a poor approximation of the perceived difference between images, except when the PSNR values are relatively high. In general, MSE-like measures do not anticipate human visual characteristics. For example, if one compares an image with a version of itself spatially shifted by a couple of pixels, the resulting MSE number will be high, even though the images are almost indistinguishable. Also, MSE or PSNR are not commonly applied to color images. Nevertheless, if one compares compression methods yielding high PSNR (e.g., above 30 to 35 dB), an advantage of 1 dB typically implies noticeably better image quality.

The perfect distortion measure still eludes researchers and is the subject of intense debates (see Reference 13, Chapters 11 through 15.) Another measure of distortion better suited to color images is CIE's ΔE[14,15] and its extension, the so-called spatial ΔE ($S\Delta E$).[16] In the $S\Delta E$, the image is mapped into a device-independent CIELAB color space (and, optionally, farther into a linear luminance-chrominance color space). The luminance is filtered using a spatial filter that provides a linear shift invariant approximation to the spatial response of the human visual system for luminance signals. Another filter is applied to the chrominance channels, where the filter was properly designed for approximating the HVS chrominance sensitivity. The average CIE ΔE between the filtered images results in the final $S\Delta E$ number. The whole process is illustrated in Figure 8.7.

8.3 Standard image coders

So far, we have not explained image compression methods in detail. There are two good reasons for this. First, the compression field is so vast that it is impossible to cover it here with reasonable depth. Second, there are a few standard image compression systems that are widely used. The inner workings of common compression engines are well known and available to many implementers. Hence, one can buy them off the shelf, and we can treat them as black boxes that are used to compress a number of monochrome images. The reader is referred to Figure 8.1c for a diagram of a typical color image compression system. After a pixel-by-pixel color transform, each transformed image plane is fed to monochrome compressors whose compressed bitstreams are multiplexed somehow to produce the output compressed stream. The color spaces and color transformations are often application

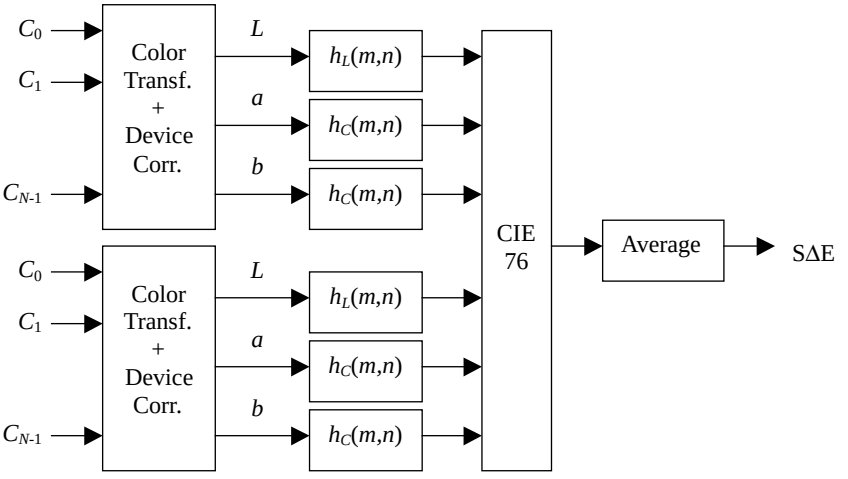


Figure 8.7 Distance between two color images for computing distortion: spatial ΔE. The color is converted to device-independent CIELAB. The channels are spatially filtered using separate filters for either luminance or chrominance planes. The average ΔE between filtered CIELAB images can be used as a distortion measure.

dependent, and we will devote the future sections to the discussion of color transforms and the nature of the resulting color planes. For this discussion, in this chapter, the reader may take these standard compression schemes at face value and treat them almost as black boxes.

The most popular image compression scheme is the JPEG standard.[17] Details along with the standard's text can be easily found in the literature.[17] JPEG compresses up to four color planes independently, and details such as color spaces are left to the application level. Application information can be conveyed to the receiver through so-called application markers. Interestingly enough, JPEG is highly popular because of its publicly available implementation developed by the Independent JPEG Group (IJG).[18] In the absence of any standard method for describing color, the IJG defined the "JFIF" application marker. When present, among other things, it indicates to the decoder that the color space was YCbCr, which is a simple linear transform of the input RGB data that will be described later.

JPEG encodes each color plane following the steps in Figure 8.1. The image is broken into blocks of 8×8 pixels, and each block is transformed using the DCT, the transformed samples being uniformly quantized. Each of the 64 quantizers is governed by the entries of the "quantizer table," which contains the quantization steps and is the main degree of freedom in JPEG compression. The quantized data of a block are scanned following a zig-zag path as shown in Figure 8.8. The vector is input to an entropy coder, which combines run-length counting and variable length coding of the counts. The DCT samples are supposed to be small for high frequencies, and most are

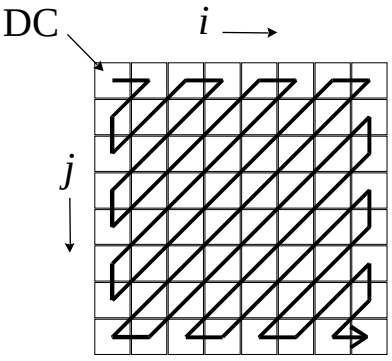


Figure 8.8 Zig-zag scanning of the DCT samples of a JPEG block of pixels into a one-dimensional vector.

set to zero (discarded) by the quantization process. Effectively, most samples are zero at the end of the zig-zag path, and this fact is explored for compression. In essence, the more zero-value quantized samples there are in the zig-zag scanned vector, the higher the compression. A better introduction and complete description should be sought elsewhere.[17] For now, we just need to think of it as a coder that processes each color plane independently. JPEG is available in virtually all software that involves digital images.

Another compression system that has recently been approved as an international standard is the JPEG 2000.[19–21] JPEG 2000 is meant as an improvement on DCT-based JPEG, but, instead of solely increasing the compression efficiency, it provides increased functionality via new compression paradigms. Conceptually, JPEG 2000 is a transform coder that is based on the steps depicted in Figure 8.1. It employs a wavelet transform and uniform quantization. Commonly, the steps sizes are very small, which, in effect, makes the quantization an integer representation of the wavelet transformed data. The data can be represented in binary code so that the transformed coefficients can be seen as formed by bit planes as illustrated in Figure 8.9.

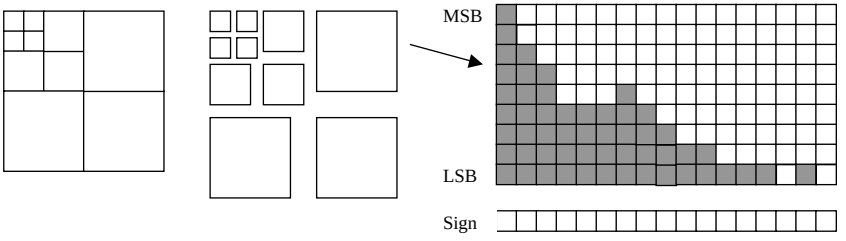


Figure 8.9 Each sub-band of the wavelet transform is represented via the sign and magnitude of each of its coefficients, thus forming bit planes. Each bit plane of each sub-band is compressed using arithmetic coding.

In essence, the most significant bit plane is encoded first, followed by the other planes. Compression is achieved by applying arithmetic coders to compress the bit planes, using contextual modeling of the probabilities.[19–21] Compressed data for each bit plane, and for each sub-band, can be encapsulated separately. As a result, data for reconstructing the image can be progressively sent to the receiver. The progression can be either by resolution (sending all bit planes for one sub-band level before proceeding to the next higher resolution level) or by quality (sending one bit plane information for all sub-bands before proceeding to the next bit plane). The process is illustrated in Figure 8.10.

JPEG 2000 was meant from its conception to be feature rich. There are many other features in JPEG 2000. The reader is encouraged to read the standard itself, or one of the many papers on the JPEG 2000 effort, to appreciate its full feature set.[19–21] For our purposes, it suffices to say that it is an efficient state-of-the-art wavelet-based compression method that can be used to compress a multitude of color planes. JPEG 2000 provides for a few color transformations. Most interesting is the reversible color transform, which will be discussed later.

Another important compression method is the JPEG-LS standard.[22,23] It is a low-complexity compressor aimed at lossless or near-lossless image compression. JPEG-LS is based on pixel prediction and prediction error encoding (a very sophisticated DPCM coder, in essence). Prediction is adaptive, and compression is based on variable length coding.

All these JPEG coders were devised to fulfill different purposes and to address different image compression needs, but all of them treat each color image channel independently. Hence, for the following discussion, it would be sufficient to assume the compression will be performed using any one of these compressors (JPEG, JPEG 2000, or JPEG-LS).

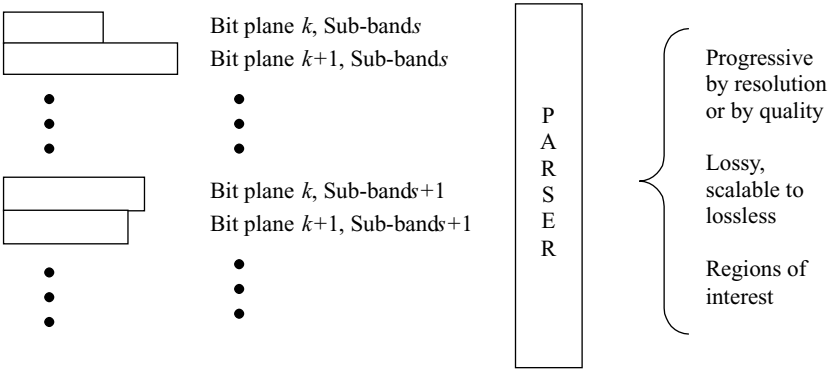


Figure 8.10 The data for each bit plane and for each sub-band are packetized. The compressor can write the file once, and the transmitter or decoder can decide how to read the data by parsing and gathering the compressed data in any suitable order.

8.4 Multidimensional color model and transforms

A color image is represented as a finite number of color image planes. Each color is obtained by filtering the image (pixel) spectrum and by measuring the resulting luminosity energy. In this way, each pixel color is represented by a few values, corresponding to a few filters. Usually, three filters (RGB) are used, but multispectral data as well as subtractive spaces such as CMYK use more than three channels. A sampled color image is an array of N-tuples: every pixel is a vector. For a continuous (not spatially sampled) image, another interpretation is that an image is a Riemannian surface in $(N + 2)$-D space. For the typical three-channel case, the image is a five-dimensional parametric surface where the parameters are chosen as the spatial coordinates xy. One and only one point is mapped to each point in the xy plane. To see this, imagine a scan line of the image (one-dimensional signal) and one single-color signal (monochrome). Then, the "image" is a simple function as in Figure 8.3c. If the image is a single -two-dimensional plane (monochrome), it is a surface; i.e., it is parameterized in two-dimensions as illustrated in Figure 8.11a and 8.11b. To add more than one color signal, imagine a one-dimensional signal (scanline) and two color signals. The "image" would be a line in three-dimensions as depicted in Figure 8.11c. Extending the space to two-dimensions and the number of color channels to three, it is easy to conclude that a three-color image is a surface in five-dimensional parameterized by two out of five axes. This abstraction is useful to re-emphasize that there is correlation not only across color planes or across space within a color plane, but also across both color planes and space coordinates. In other words, correlation exists within a five-dimensional space! Furthermore, this simple topological representation is linear.

Of interest to us are the properties of color images related to compression. In particular, well-correlated signals or regions without much detail tend to compress better than more noisy or detailed images; i.e., smooth regions compress better than regions involving too many edges.

Let $C_k(m, n)$ denote the kth color plane at pixel position (m, n), and let $\mathbf{c}(m, n)$ be a vector containing all such color planes; i.e., $\mathbf{c}^T(m, n) = [C_0(m, n), C_1(m, n), \ldots, C_{N-1}(m, n)]$. Let us form a composite vector

$$\mathbf{u}^T = [\mathbf{c}^T(m,n), \ldots, \mathbf{c}^T(m + k_1, n + k_2), \ldots] \qquad (k_1, k_2) \in \Xi \tag{8.9}$$

where Ξ is a set determining a neighborhood of K pixels around the origin. A measure of correlation among neighbor pixels across color planes can be made via the autocorrelation matrix of $\mathbf{u}$, i.e.,

$$\mathbf{R}_u = E\{\mathbf{u}\mathbf{u}^T\} = \begin{bmatrix} \Gamma_0 & \Gamma_1 & \ldots & \Gamma_K \\ \Gamma_1 & \Gamma_0 & & \\ \vdots & & \ddots & \\ \Gamma_K & & & \Gamma_0 \end{bmatrix} \tag{8.10}$$

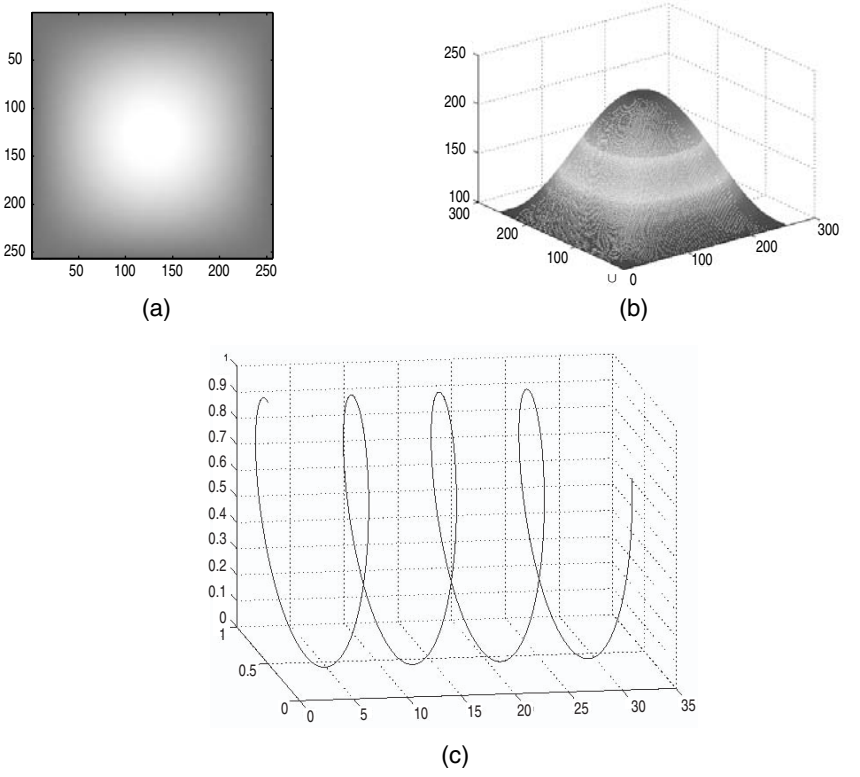


Figure 8.11 Geometric representation of color images (a) a monochrome image displayed as light intensity, (b) the same image represented as a surface in three-dimensions, and (c) a one-dimensional image with two color channels represented as a parametric line in three dimensions.

where Γ_0 is the plain correlation across color planes, while Γ_K represents the correlation of color values over spatially displaced pixels.

To decorrelate the samples of $\mathbf{u}$, a transformation $\mathbf{A}$ such that $\mathbf{y} = \mathbf{A}\,\mathbf{u}$ can be used to decorrelate the data. The KLT we mentioned earlier is the one such that

$$\mathbf{R}_y = \mathrm{E}\{\mathbf{y}\mathbf{y}^T\} = \mathbf{A}\,\mathbf{R}_u\,\mathbf{A}^T \tag{8.11}$$

is a diagonal matrix.[2,4,5,7,9] This can be accomplished by choosing the rows of $\mathbf{A}$ as being the eigenvectors of $\mathbf{R}_u$. Note that the transform $\mathbf{A}$ will simultaneously decorrelate the image within and across color planes. In other words, it achieves both color and space decorrelation.

The following example will illustrate the process. Assume the immediate horizontal, vertical, and diagonal neighbors so that $\Gamma_h = E\{\mathbf{c}(m, n)\mathbf{c}^T(m + 1, n)\}$, $\Gamma_v = E\{\mathbf{c}(m, n)\mathbf{c}^T(m, n + 1)\}$ $\Gamma_d = E\{\mathbf{c}(m, n)\mathbf{c}(m + 1, n + 1)\}$. For the RGB image shown in Figure 8.12, $\mathbf{u}^T = [r(m, n), g(m, n), b(m, n), r(m, n + 1), \ldots,$

Figure 8.12 Color (RGB) channels of the image used for the multidimensional transform example.

$g(m+1, n+1)$, $b(m+1, n+1)$] and the following correlation matrices were found:

$$\Gamma_0 = \begin{bmatrix} 0.9757 & 0.9551 & 0.8661 \\ 0.9551 & 1.0000 & 0.9242 \\ 0.8661 & 0.9242 & 0.9185 \end{bmatrix} \quad \Gamma_h = \begin{bmatrix} 0.9392 & 0.9195 & 0.8346 \\ 0.9195 & 0.9630 & 0.8912 \\ 0.8346 & 0.8912 & 0.8862 \end{bmatrix}$$

$$\Gamma_v = \begin{bmatrix} 0.9295 & 0.9073 & 0.8212 \\ 0.9073 & 0.9521 & 0.8788 \\ 0.8212 & 0.8788 & 0.8749 \end{bmatrix} \quad \Gamma_d = \begin{bmatrix} 0.9096 & 0.8889 & 0.8059 \\ 0.8889 & 0.9329 & 0.8624 \\ 0.8059 & 0.8624 & 0.8591 \end{bmatrix} \tag{8.12}$$

As expected, the within-pixel cross correlation is the largest, followed by the correlation among planes of horizontal or vertical neighbor pixels. What is less expected is the large correlation among samples of different planes and pixels. The matrix **A** that diagonalizes $\mathbf{R}_u$ is shown in Figure 8.13, while the standard deviations of the transformed signals are

$$[\text{diag}(\mathbf{R}_y)]^{1/2} = [1.0000\ 0.1716\ 0.1448\ 0.1196\ 0.0788\ 0.0684\ \ldots \\ \ldots\ 0.0221\ 0.0116\ 0.0113\ 0.0071\ 0.0050\ 0.0014] \tag{8.13}$$

0.2888	0.2974	0.2796	0.2888	0.2974	0.2796	0.2888	0.2974	0.2796	0.2888	0.2974	0.2796
-0.3490	0.0022	0.3580	-0.3490	0.0022	0.3580	-0.3490	0.0022	0.3580	-0.3490	0.0022	0.3580
0.2913	0.2951	0.2794	0.2913	0.2951	0.2794	-0.2913	-0.2951	-0.2794	-0.2913	-0.2951	-0.2794
-0.2953	-0.2995	-0.2704	0.2953	0.2995	0.2704	-0.2953	-0.2995	-0.2704	0.2953	0.2995	0.2704
0.2117	-0.4020	0.2089	0.2117	-0.4020	0.2089	0.2117	-0.4020	0.2089	0.2117	-0.4020	0.2089
-0.2868	-0.2961	-0.2830	0.2868	0.2961	0.2830	0.2868	0.2961	0.2830	-0.2868	-0.2961	-0.2830
0.3372	0.0008	-0.3691	-0.3372	-0.0008	0.3691	0.3372	0.0008	-0.3691	-0.3372	-0.0008	0.3691
-0.2215	0.4004	-0.2015	0.2215	-0.4004	0.2015	-0.2215	0.4004	-0.2015	0.2215	-0.4004	0.2015
-0.3739	0.3292	0.0422	-0.3739	0.3292	0.0422	0.3739	-0.3292	-0.0422	0.3739	-0.3292	-0.0422
-0.2014	-0.1988	0.4122	0.2014	0.1988	-0.4122	0.2014	0.1988	-0.4122	-0.2014	-0.1988	0.4122
0.1591	0.2336	-0.4125	0.1591	0.2336	-0.4125	-0.1591	-0.2336	0.4125	-0.1591	-0.2336	0.4125
0.3566	-0.3504	0.0052	-0.3566	0.3504	-0.0052	-0.3566	0.3504	-0.0052	0.3566	-0.3504	0.0052

Figure 8.13 Transformation matrix **A** for the example image.

Note the rapid decay of the energy of the transformed samples, i.e., the high energy compaction, which is often a sign of high compression. The multidimensional transform will outperform the compaction provided by separate color transforms followed by linear spatial transforms (such as DCT) of the same sizes. Despite the theoretical advantage, these transforms are not commonly known or used. They are image dependent, and some image-independent separate transforms also provide reasonable performance, as we will discuss later.

8.5 Color transforms

Often, the spatial and color transforms are independent, as depicted in Figure 8.1c. Each pixel is first transformed separately to remove color redundancy, then a spatial transform is applied to reduce the spatial redundancy.

If the color transform is a linear matrix **Q**, the within-pixel color transform process is reduced to constraining the space–color transform into a block diagonal matrix,

$$\mathbf{A} = \text{diag}(\mathbf{Q}, \mathbf{Q}, \ldots, \mathbf{Q}) \tag{8.14}$$

The reason for using color transforms is to enhance the compression performance. As we discussed, compression is improved if we reduce both rate and distortion or a cost function that is a linear combination of both. Typically "smooth" images (i.e., those lacking too many details and sharp edges) are more easily compressed than textured and detailed images. By "more easily," we mean achieving higher compression for the same distortion or less distortion for the same compression. How do we chose **Q** in Equations 8.11 and 8.14 so as to favor compression? It can be shown that, if the transform has the form of Equation 8.14, and if we make $\mathbf{R}_y = \mathbf{B}\otimes\mathbf{D}$, we can optimize the *RD* trade-off, given some mild conditions, where **B** is some Toeplitz matrix, **D** is a diagonal matrix, and ⊗ denotes the Kronecker product.

Decorrelation of the color planes without spatial considerations is achieved by the pixel-wise Karhunen-Loeve transform (KLT).[4,5] To find the KLT, one just uses $\Xi = \{(0, 0)\}$ in Equation 8.9, or $\mathbf{u} = \mathbf{c}(m, n)$, so that $\mathbf{R}_u = \Gamma_0$ in Equation 8.10, and $\mathbf{A} = \mathbf{Q}$ is selected as the matrix containing the eigenvectors of $\mathbf{R}_u = \Gamma_0$. The KLT approach is general and should provide good performance for any color space, including multispectral data.

The disadvantage of the KLT is the fact that one needs to gather the statistics of the image ($\mathbf{R}_u$). However, some useful transforms provide reasonable plane decorrelation for most typical images. Color spaces such as YIQ, YUV, and YES are simple linear transformations of linear RGB planes. In fact, a very important (perhaps the most used) color transformation is the one that brings RGB into YCbCr. YCbCr is a variant of YUV and is defined by the following matrix transformation:

$$\mathbf{Q}_{YCC} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.168 & -0.332 & 0.5 \\ 0.5 & -0.418 & -0.082 \end{bmatrix} \qquad \mathbf{Q}_{YCC}^{-1} = \begin{bmatrix} 1 & 0 & 1.402 \\ 1 & -0.344 & 0.714 \\ 1 & 1.772 & 0 \end{bmatrix} \tag{8.15}$$

In the JPEG 2000 jargon, the above transform is also referred to as an irreversible color transform (ICT),[19–21] as it uses floating point numbers, and the YCbCr samples need to be re-quantized. However, in JPEG 2000, one can use reversible wavelets that would allow lossless compression. All the efforts to provide lossy-to-lossless scalability in the JPEG 2000 architecture would be in vain if the ICT was used. The JPEG 2000 committee decided to approve an approximation of ICT as an optional transform that would allow total reversibility while providing reasonable decorrelation. Such an approximation is the reversible color transform (RCT), which is defined by the following formulae:[19]

$$\begin{aligned} Y_0 &= \left\lfloor \frac{C_0 + 2(C_1 + C_2)}{4} \right\rfloor \\ Y_1 &= C_2 - C_1 \\ Y_2 &= C_0 - C_1 \end{aligned} \tag{8.16}$$

where Y_0, Y_1, Y_2 are the transformed color planes, and $\lfloor \cdot \rfloor$ denotes the "floor" operator; i.e., discard the decimal places. The original color planes can be perfectly reconstructed from an integer representation of Y_0, Y_1, Y_2 as

$$\begin{aligned} C_1 &= Y_0 - \left\lfloor \frac{Y_1 + Y_2}{4} \right\rfloor \\ C_0 &= Y_2 + C_1 \\ C_2 &= Y_1 + C_1 \end{aligned} \tag{8.17}$$

Note that Y_1 and Y_2 require one more precision bit than Y_0.

Other important color spaces for compression are CIELAB and CIELUV, which are covered elsewhere in this handbook. Color fax systems demand compression of images in a CIELAB color space.[24] In all these cases, the conversion takes RGB data into some luminance–chrominance color space. An important aspect of these color spaces is that the human eye has lower sensitivity to high-frequency components (details) of the chrominance images. Hence, it is easy to subsample, or rather compress, more aggressively the chrominance components.

Other noteworthy transformations are those pertaining to the compression of CMYK image data.[25–27] JPEG's SPIFF file header specification[28] defines the YCbCrK color space as a derivation of CMYK data. Let us start with the negative (inverse) of CMY, i.e., $C_0C_1C_2C_3$ are indeed RGBK. Then, the transformation from RGBK (CMYK) to YCbCrK is simply

$$\mathbf{Q}_{YCCK} = \begin{bmatrix} & \mathbf{Q}_{YCC} & \begin{matrix}0\\0\\0\end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8.18}$$

An improved version of the above transformation is the one that brings the data into the Y_+Y_CbCr or YYCC color space. If we invert all CMYK channels so that $C_0C_1C_2C_3$ becomes RGBW, where W (white) is just the inverse of the K (black or key) channel, then the transform is defined as

$$\mathbf{Q}_{YYCC} = \begin{bmatrix} 1/2 & 0 & 0 & 1/2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1/2 & 0 & 0 & -1/2 \end{bmatrix} \begin{bmatrix} & \mathbf{Q}_{YCC} & \begin{matrix}0\\0\\0\end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8.19}$$

which can be implemented as in the flow graph depicted in Figure 8.14. Note that CMYK planes are often device-specific data. Simple linear transformations of CMYK data will only aim to compact the data better, while remaining oriented to a particular device.

Several devices (e.g., printers with more than four inks) use additional colorants, for example, by adding orange colorants to the CMYK set. Multispectral data is also relevant and may use many color planes. For these cases, there is no common transform method. The KLT would work for all cases, but sometimes the image statistics are unavailable. There are some proposals of using conventional transforms such as DCT or wavelets to transform the

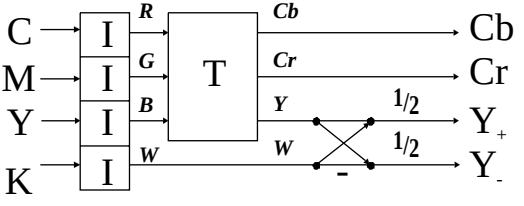


Figure 8.14 Color transform implementation from CMYK to Y_+Y_CbCr (or YYCC), where I means inversion (negative).

samples across color planes. Nevertheless, their efficacy is still being studied and naturally varies with each case.

8.6 Compressing RGB images

As we discussed, transforms that aim to decorrelate the color planes can also produce planes that more easily compressed. Figure 8.15 shows an example of three color planes before and after color transformation to YCbCr. Note how the CbCr channels are much smoother than the RGB channels. If we go back to the image planes shown in Figure 8.12, apply a transformation to YCbCr, and then remeasure the correlation as in Equation 8.12, we obtain

$$\Gamma_0 = \begin{bmatrix} 1.0000 & 0.1140 & 0.0204 \\ 0.1140 & 0.0559 & 0.0239 \\ 0.0204 & 0.0239 & 0.0739 \end{bmatrix} \quad \Gamma_h = \begin{bmatrix} 0.9424 & 0.1103 & 0.0196 \\ 0.1103 & 0.0541 & 0.0232 \\ 0.0196 & 0.0232 & 0.0715 \end{bmatrix}$$

$$\Gamma_v = \begin{bmatrix} 0.9281 & 0.1116 & 0.0232 \\ 0.1116 & 0.0539 & 0.0234 \\ 0.0232 & 0.0234 & 0.0712 \end{bmatrix} \quad \Gamma_d = \begin{bmatrix} 0.9030 & 0.1089 & 0.0220 \\ 0.1089 & 0.0531 & 0.0231 \\ 0.0220 & 0.0231 & 0.0804 \end{bmatrix} \tag{8.20}$$

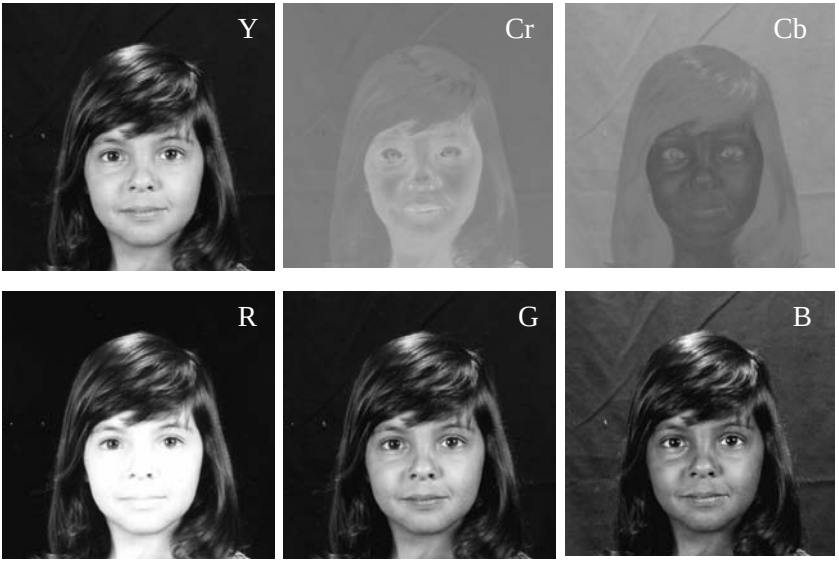


Figure 8.15 YCbCr color channels compared to the original RGB ones. Note the lack of details in the chrominance (CbCr) channels.

Note how decorrelated the planes are as compared to the RGB channels in Equation 8.12. The cross correlations are largely reduced, and the energy is concentrated in one channel. If we model the image planes as a first-order Markov process with correlation coefficient ρ, in this example, this coefficient is about 0.92 to 0.94 for the luminance channel and about 0.96 to 0.97 for the chrominance ones. This fact is additional evidence that the chrominance channels are typically "smoother" than luminance channels. Similar conclusions can be reached by inspecting the Fourier transforms of the resulting YCbCr color planes.

It is clear that the transformation RGB–YCbCr produces "compression-friendly" color planes. The question is, how do we compare this space with, for example, device-independent CIELAB? To answer this question, RD plots are shown in Figure 8.16, comparing JPEG compression using both CIELAB and YCbCr. In these experiments, default (example) luminance and chrominance tables were used, and distortion is given as both PSNR and $S\Delta E$. Also, the RD curves in Figure 8.16 were obtained by varying a scaling parameter for quantizer tables, luminance, and chrominance. This scaling is also known as the *quality* factor and gives one knob to regulate compression; i.e., a single parameter yields the RD points, hence a curve in RD space. The plots shown in Figure 8.16 are averages over several images. For same rate, YCbCr typically yields higher PSNR or lower $S\Delta E$. Conversely, for the same distortion, YCbCr typically demands less rate than CIELAB. The plots in Figure 8.16 are typical and serve to illustrate that it is commonly advantageous to compress RGB images using the transformation to YCbCr instead of compressing under CIELAB. The exception is for very low bit rates, where CIELAB becomes more competitive.

A typical JPEG (or JPEG 2000) compressor implementation would treat the luminance channel differently from the chrominance ones. The differentiation can be to employ different quantizer tables and to subsample the chrominance planes. Often, CbCr will be reduced by a factor of two in each

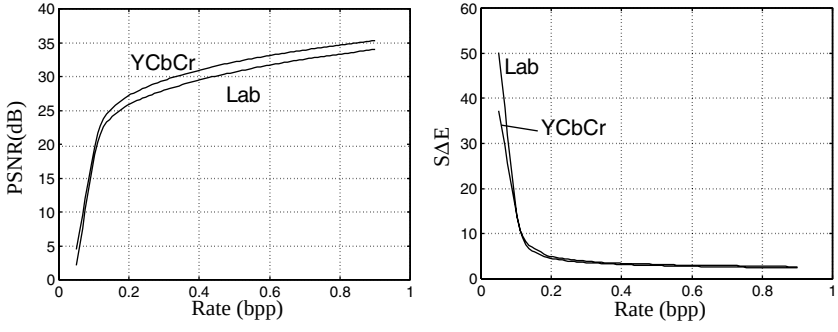


Figure 8.16 RD plots comparing compression using YCbCr and CIELAB. Results show average distortion for several images using the JPEG coder. The YCbCr space is typically slightly superior to CIELAB, except for very low bit rates.

direction before compression. The claim is that, because we are less sensitive to the high frequency of chrominance components, one could reduce the image planes from the start without loss of visual fidelity. The problem with that argument is that, by reducing the data, one increases distortion and compression. If one does not reduce the chrominance planes but instead relies on the increasing quantizer steps, one would also increase distortion and compression. The question is, which one is a better trade-off? Because of subsampling, data are irreversibly lost no matter how high the bit rate. So, for high enough bit rates, it is better not to subsample the planes. For low rates, subsampling artifacts might be better than compression artifacts. So, the best approach may actually depend on the bit-rate target. There is a breakpoint at which the curves with and without subsampling would cross. To clarify this issue, RD plots comparing the performance of a JPEG coder with and without chrominance (Cb and Cr) subsampling are shown in Figure 8.17. The plots were obtained in the same conditions described for Figure 8.16. For rates above a certain breakpoint, it is always better not to subsample the CbCr planes. This breakpoint is commonly around 0.2 bpp or at a compression ratio about 120:1 (starting with the original 24-bpp RGB image).

It is safe to say that, for the average compression of RGB images, one would be better off using the YCbCr transformation and not subsampling chrominance planes.

8.7 Compressing CMYK images

CMYK data is targeted for a particular device. The correlation between K and the other channels can change drastically from device to device. A color transform for compressing CMYK images likely would not work across all devices. For that reason, we would be content with a "good" solution that works across most devices, as the alternative is a case-by-case study.

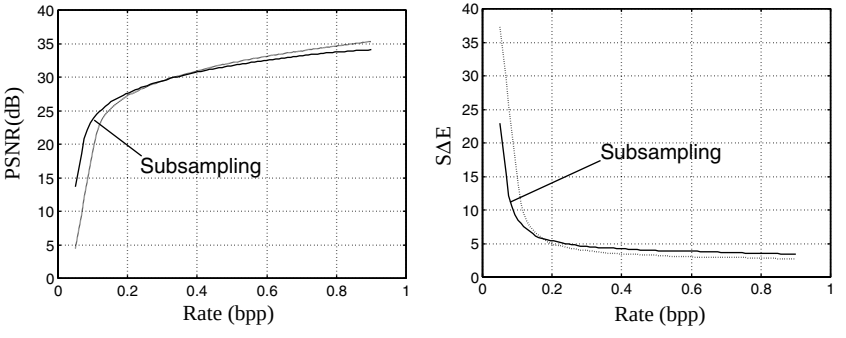


Figure 8.17 RD plots comparing the performance of a JPEG coder with and without chrominance subsampling. For rates above a certain breakpoint, it is always better not to subsample the CbCr (or ab) planes. This breakpoint is commonly ≈0.2 bpp.

The key difference between compressing CMYK and RGB images lies in the concept of luminance–chrominance models.[25–27] Even though "luminance" and "chrominance" are derived from colorimetric principles (e.g., luminance aligned to an achromatic axis), as far as compression is concerned, what determine an image luminance are the spatial characteristics of the color planes. With three-plane images, it is easy to define luminance and chrominance. In Figure 8.15, it is natural to assign the channel that most resembles a "monochrome" version of the image as the luminance channel. However, if we replace Y channel by any of the RGB channels, i.e., an RGB to RCbCr transformation, one would likely designate R as the luminance. This is so because the "chrominance" channels are definitely distinct from what one perceives as a monochrome version of the color image. Figure 8.18a through 8.18d shows typical CMYK color planes, already inverted (i.e., 1 – C, 1 – M, 1 – Y, 1 – K, or RGBW). This particular rendering was performed for a given xerographic device. Note the large correlation between the K color plane and each of the others. That, of course, depends on the strategy for calculating K from CMY. If we use the transformation from CMYK to YYCC depicted in Figure 8.14, the resulting color planes corresponding to those at Figure 8.18a through 8.18d are shown in Figure 8.18e through 8.18h. Back to our discussion on luminance vs. chrominance, Figure 8.18e likely contains what most of us would call "luminance," whereas Figure 8.18g and 8.18h contain images that we would call "chrominance." However, it is difficult to fit the image in Figure

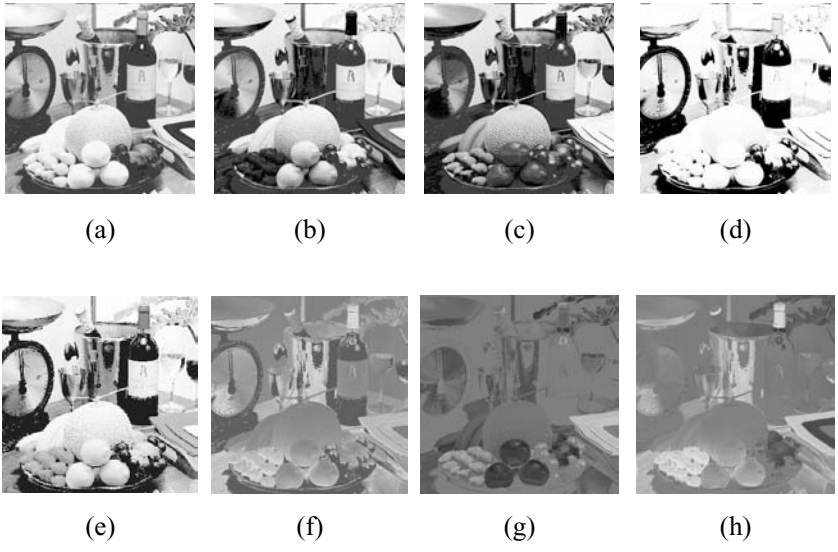


Figure 8.18 CMYK color planes for a test image. The inverses of CMYK planes are shown in (a) through (d), respectively. After the transformation to YYCC space, the color planes are (e) Y_+, (f) Y_-, (g) Cb, and (h) Cr.

8.18f to any of the "models." It contains typical luminance and chrominance spatial characteristics. This ambiguity is amplified for multispectral data where there are more channels, and any transformation of the input data will produce some with both luminance and chrominance spatial characteristics. So, why do we need to designate a color plane as luminance or chrominance anyway? It is common for compressors to have more aggressive settings for chrominance planes than for luminance. By tagging the channel as chrominance (thus, visually less important), one can exploit the benefits of more aggressive compression while being more conservative in compressing the luminance. This is a complex issue beyond the scope of this chapter. It is advisable, however, in the YYCC case to apply "chrominance" settings to two of the channels while applying "luminance" settings to the other two.

There are several transform options for the compression of CMYK data, including

1. No transform, i.e., compress CMYK independently
2. Compress YCbCrK planes
3. Compress YYCC planes

Figure 8.19 shows RD plots for the different compression schemes for several images. Distortion is given as both $S\Delta E$ and PSNR. The plots include results with and without subsampling of the CbCr planes in the YCbCrK and YYCC schemes. YYCC typically outperforms the others for most bit rates. As in the case of compressing RGB images, chrominance subsampling is just effective for high compression ratios, i.e., ratios larger than 150:1 (less than 0.2 bpp starting with CMYK at 8 bpp each). Summarizing, YYCC space

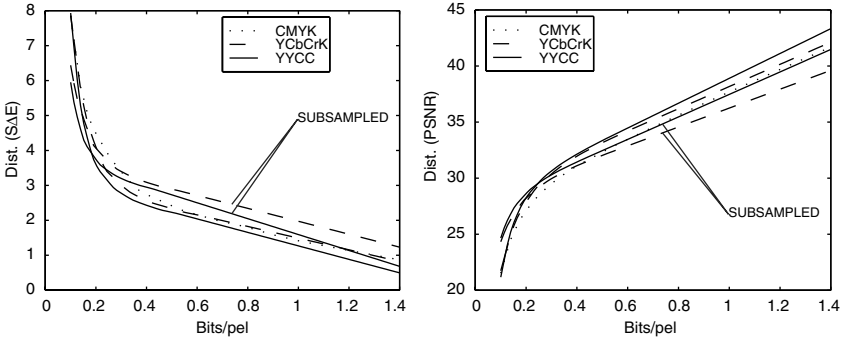


Figure 8.19 Compression performance. For a number of CMYK images rendered for different devices, the RD plots compare to the JPEG compression of the planes under three different color spaces. Distortion measure was computed as both $S\Delta E$ and PSNR. For the YCbCrK and YYCC cases, tests were performed with and without chrominance subsampling.

without chrominance subsampling seems to be a good choice for compressing CMYK data.

8.8 Summary

In this chapter, we exposed the reader to the basic aspects of the compression of digital color images. That includes basic compression concepts such as coding and quantization and the fact that compression is achieved by removing the statistical redundancy and the visual irrelevancy contained in the image. In fact, removing these two types of information determines whether one attains lossy or lossless compression. We have presented the transform coding model, along with the motivation for energy compaction image transformation in the context of compression. For that, the DCT and the wavelet transform were described.

Another model is the predictive coding model, which is popular for lossless and near-lossless compression. The key factor in devising and applying compressors to images is to understand that there is always a rate-distortion trade-off in setting up compression parameters. The best operational point lies somewhere on the LCH of the RD points. As for distortion, popular objective distortion measures are the PSNR, and $S\Delta E$ distance measures. Standard coders exist and are ready for our use, which include JPEG 2000, JPEG, and JPEG-LS. We should use them but understand the effects of their parameter choices on the image quality, along with the choice of the proper color transform.

We did not intend to describe compression systems in detail, which are often designed for monochrome images. Furthermore, the popular compression systems are international standards, which are very well covered elsewhere. The focus of this chapter was on the interaction between the color image representation and existing compression systems.

A multidimensional color model was discussed to show that there might be strong correlation within and across pixels, simultaneously. We have shown simple color and spatial correlation measurements. Even though there is a multidimensional correlation, most of the compression systems apply color transforms independently from the spatial transform. Popular color transforms for compression were discussed in detail. That includes the transformation from RGB to YCbCR and the KLT. The YCbCr transformation is the irreversible color transform. This, and the reversible color transform, make up the main color transform options for JPEG 2000. Apart from the KLT, other color transforms for non-RGB data were discussed, including the transformations from CMYK to YCbCrK and to YYCC. The concepts of luminance vs. chrominance were discussed with respect to the compression settings. Compression of RGB and CMYK images was discussed, comparing color transforms and other settings.

We aimed at presenting a few basic compression concepts applied to color images. We regard this chapter as an introduction to the subject so, rather than serving as a thorough reference, we hope this chapter will inspire

the reader to explore the subject further in the following references and through independent experimentation.

References

1. Rao, K. R. and Hwang, J., Techniques and standards for image, in *Video and Audio Coding,* Prentice-Hall, Upper Saddle River, NJ, 1996.
2. Netravali, A. and Haskell, B., *Digital Pictures: Representation and Compression,* Plenum Press, New York, 1988.
3. Storer, J. A., Ed., *Image and Text Compression,* Kluwer Academic, Norwell, MA, 1992.
4. Pratt, W., *Digital Image Processing,* John Wiley & Sons, New York, 1978.
5. Rao, K. R. and Yip P., Eds., *The Transform and Data Compression Handbook,* CRC Press, Boca Raton, FL, 2001.
6. Rabbani, M. and Jones P., *Digital Image Compression Techniques,* SPIE Press, Bellingham, WA, 1991.
7. Gersho, A. and Gray R., *Vector Quantization and Signal Compression,* Kluwer Academic, Norwell, MA, 1992.
8. Gallager, R. G., *Information Theory and Reliable Communication,* John Wiley & Sons, New York, 1968.
9. Rao, K. R. and Yip, P., *Discrete Cosine Transform, Algorithms, Advantages and Applications,* Academic Press, San Diego, CA, 1990.
10. Vetterli, M. and Kovacevic, J., *Wavelets and Subband Coding,* Prentice-Hall, Englewood Cliffs, NJ, 1995.
11. Strang, G. and Nguyen, T., *Wavelets and Filter Banks,* Wellesley-Cambridge, Wellesley, MA, 1996.
12. Gray, R., *Source Coding Theory,* Kluwer Academic, Norwell, MA, 1993.
13. Watson, A. B., Ed., *Digital Images and Human Vision,* MIT Press, Cambridge, MA, 1993.
14. CIE Publication No. 15.2, *Colorimetry,* Bureau Central de la CIE, Vienna, 1986.
15. Robertson, A. R., Historical development of CIE recommended color difference equations, *Color Res. Appl.,* 15(3), 167–170, 1990.
16. Zhang, X. M. and Wandell, B. A., A spatial extension to CIELAB for digital color image reproduction, in *Proc. Soc. for Info. Display Symp.,* 1996.
17. Pennebaker, W. B. and Mitchell, J. L., *JPEG: Still Image Compression Standard,* Van Nostrand Reinhold, New York, 1993.
18. Independent JPEG Group Library, http://www.ijg.org.
19. Taubman, D. and Marcellin, M., *Jpeg 2000: Image Compression Fundamentals, Standards, and Practice*, Kluwer Academic Press, Dordrecht, the Netherlands, 2001.
20. Marcellin, M., Gormish, M. J., Bilgin, A., and Boliek, M., An overview of JPEG-2000, in *Proc. 2000 Data Compression Conference,* Snowbird, Utah, March 2000.
21. Christopoulos, C., Skodras, A., and Ebrahimi, T., The JPEG 2000 still image coding system: an overview, *IEEE Trans. Consumer Electronics,* 46(4), 1103–1127, 2000.
22. ISO/IEC FCD 14495-1 — Lossless and near-lossless compression of continuous-tone still images, http://www.jpeg.org/public/fcd14495p.pdf, 1997.
23. Weinberger, M., Seroussi, G., and Sapiro, G., From LOCO-I to JPEG-LS standard, in *Proc. Int. Conf. Image Proc. (ICIP'99),* 24A01.7, Kobe, Japan, 1999.

24. Buckley, R., Venable, D., and McIntyre, L., New developments in color facsimile and internet fax, in *Proc. IS&T's Fifth Color Imaging Conference,* Scottsdale, AZ, November 1997, 296–300.
25. de Queiroz, R., On independent color space transformations for the compression of CMYK images, *IEEE Trans. Image Processing,* 8, 1446–1451, 1999.
26. Van Assche, S., Denecker, K., and De Neve, P., Evaluation of lossless compression techniques for high-resolution RGB and CMYK color images, *J. Electronic Imaging,* 8, 415–421, 1999.
27. Van Assche, S., Denecker, K., Philips, W., and Lemahieu, I., A comparison of lossless compression techniques for prepress color images, in *IS&T SPIE Symp. on Electronic Imaging: Visual Communications and Image Processing,* in *Proc. SPIE,* 3653, San Jose, CA, January 1999, 1376–1383.
28. ISO/IEC CD 10918-3, *Info. Technology — Digital Compression and Coding of Continuous Tone Still Images — Part 3: Extensions,* November 13, 1994.

chapter nine

Color quantization

Luc Brun
Université de Reims Champagne Ardenne
Alain Trémeau
Université Jean Monnet de Saint-Etienne

Contents

9.1 Introduction

Color image quantization is used to reduce the number of colors in a digital image with a minimal visual distortion. Color quantization can also be defined as a lossy image compression operation. Until lately, quantization was used to reproduce 24-bit images on graphics hardware with a limited number of simultaneous colors (e.g., frame buffer displays with 4- or 8-bit colormaps). Even though 24-bit graphics hardware is becoming more common, color quantization still maintains its practical value. It lessens space requirements for storage of image data and reduces transmission bandwidth requirements in multimedia applications.

Given a color image I, let us denote by C_I the set of its colors and by M the cardinality of C_I. The quantization of I into K colors, with $K < M$ (and usually $K << M$) consists in selecting a set of K *representative colors* and replacing the color of each pixel of the original image by a suitable representative color.

Because the first applications of quantization were used to display full-color images on low-cost color output devices, quantization algorithms from the beginning had to face to two constraints. On one hand, the quantized image must be computed at the time the image is displayed. This makes computational efficiency of critical importance. On the other hand, the visual distortion between the original image and the reproduced one has to be as small as possible. The trade-off between computational times and quantized image quality is application dependent, and many quantization methods have been designed according to various constraints on this trade-off.

One straightforward way to obtain quantized images with low computational times consists of using a preselected set of representative colors.[29,49] Such methods, described in Section 9.2, are referenced as *image-independent quantization* methods (see also arrow 1 in Figure 9.1). Quantized images with higher quality are generally obtained by building the set of representative colors according to the color distribution of the input image. Such methods are called *adaptive quantization* methods or *image-dependent quantization* methods (arrow 2 in Figure 9.1).

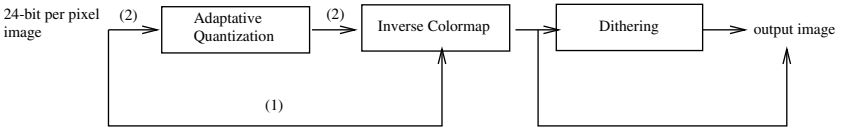


Figure 9.1 The sequence of algorithms applied to produce the output image.

Given an input image and a set of representative colors, the mapping of each color of the original image to a representative one is performed by the *inverse colormap operation* (Figure 9.1 and Section 9.7). Because each color of the original image is mapped onto a representative color, the definition of the set of representative colors induces a partition of the image color set C_I. The notions of color set partition and representative colors are thus closely linked, and many quantization methods define first a partition of C_I into a set of clusters and induce from it a set of representative colors. The brute force of enumerating all possible partitions of the set C_I with M colors into K subsets is out of the question here, as the number of all possible partitions,[5,75]

$$\frac{1}{K!}\sum_{k=0}^{K}(-1)^{K-k}\binom{K}{k}k^M$$

is astronomical for even very small values of M and K.

The definition of a partition of C_I is thus achieved by different heuristics described throughout this chapter (see also Reference 58 for an overview of digital color imaging). Each heuristic is designed according to some constraints on processing times and a particular definition of the visual distortion between the input and output images. Note that this notion of visual distortion may also be application dependent.

One other family of quantization methods first defines an initial set of representatives and improves it iteratively by using the partition of C_I induced by the set of representatives. Such methods are referenced as *post-clustering methods* and are closely linked to the tracking of function minima by iterative methods.

The optimal goal of adaptive quantization methods is thus to build a set of representative colors such that the perceived difference between the original image and the quantized one is as small as possible. The definition of relevant criteria to characterize the perceived image quality[62,63,65] is still an open problem. The difficulty of this problem is reinforced by our limited knowledge of the human visual system (HVS). There is thus no universal criterion available to characterize the perceived image similarities. One criterion commonly used by quantization algorithms is minimization of the distance between each input color and its representative. Such criteria may be measured, for instance, using the total squared error (Section 9.4), which minimizes the distance within each cluster. A dual approach attempts to maximize the distance between clusters (Section 9.4.3). Note that the distance

from each color to its representative is relative to the color space in which the total squared error is computed. The choice of one color space allows us to take into account the characteristics of the human visual system encoded by this color space. The perception of the spatial arrangement of colors may also be encoded by weighting the distance of each color to its representative according to the local spatial arrangement of colors (Section 9.5). This idea has been developed recently by several quantization methods (Section 9.8.4) that simultaneously optimize the selection of the representative colors and their spatial arrangements.

The spatial arrangement of representative colors may also be optimized by a post-processing step called *dithering* (Section 9.8). This last step reduces visual artifacts such as false contours, which lower the output image quality noticeably.

9.2 *Image independent quantization methods*

Digitized images are generally quantized with eight bits of resolution for each color component R, G, and B. Therefore, full-color digital display systems use 24 bits to specify the color of each pixel on the screen, which can thus display 2^{24} (16.8 million) colors.

The uniform quantization of a color space Ω divides it into K equal size sub-boxes and defines the set of representative colors as the sub-box centroids. Uniform quantization techniques differ according to the geometry of the color space in which the quantization is performed. Thus, while the color space can be quantized naturally in cubical sub-boxes, the *YIQ* color space must be quantized in skewed rectangular sub-boxes; otherwise, many representative colors fall outside the *YIQ* color gamut† and hence cannot be reached by any colors in any original image (see Figure 9.2). Likewise, the

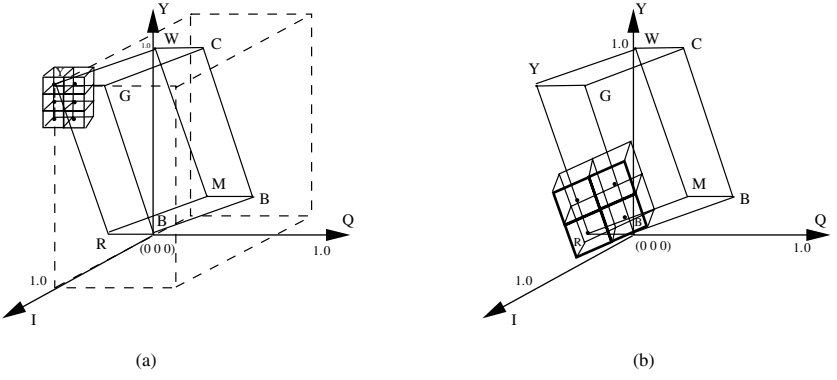


Figure 9.2 Uniform quantization of color space *YIQ* into (a) rectangular sub-boxes and (b) skewed rectangular sub-boxes.

† Color gamut refers to the range of realized colors associated with a particular device.

CIELAB color space must be linearly quantized in sub-boxes with a specific shape, even if the CIELAB color coordinates have been computed from a nonlinear transformation from the color space. The main problem of the uniform quantization of the CIELAB color space is that the exact shape of a given CIELAB color gamut is device dependent.

Image-independent color quantization techniques[49] may be viewed as a generalization of uniform quantization where the chosen quantization regions are not of the same size but are still independent of the image to be quantized. Such techniques are often used to divide a color space according to a criterion that varies throughout the color space. For example, Kurtz[42] divides the RGB color space in such a way that the distance between the representatives of any two adjacent regions are perceptually constant. Because RGB color space is not perceptually uniform, the size and shape of the region associated with each representative are relative to its location in the RGB color space.

Image-independent quantization algorithms place the K representative colors in the color gamut independently of their frequency of occurrence in a specific image, and of any other color characteristics computed from the studied image. These quantization methods are therefore computationally less expensive and simpler to implement than many image-dependent color quantization algorithms. However, using such methods, many representative colors may be assigned to locations in the color gamut where few colors of the original image reside. This is why image-dependent color quantization algorithms that use representative colors based on the image to be quantized are typically preferred.

9.3 Preprocessing steps of image-dependent quantization methods

Due to the complexity of color quantization, preprocessing steps are commonly employed for reducing the data processed by quantization algorithms. The major techniques used to reduce the data are *prequantization*, which reduces the range of each coordinate in the color space, and the *histogram calculation*, which allows us to manage the image color set more efficiently.

9.3.1 Prequantization

Prequantization, in five bits for each color component, is commonly used by many algorithms as a basic step for the quantization process. The main purpose of this prequantization is to enable a trade-off between quantizer complexity and image quality. Indeed, Fletcher[25] has shown that this initial step noticeably reduces the memory and computational time requirements of quantization algorithms. Other approaches have exploited this advantage further by using a prequantization in four bits of resolution for each color component,[37] or even in three bits of resolution for each color component.[32]

The main drawback of these approaches is that they do not take into account the nonuniformity of the human visual system to perceived color differences. Indeed, while this prequantization may produce unnoticeable degradations in high-frequency areas, it can simultaneously produce visible degradations in low-frequency areas or at the borders of contours.[61,63] Balasubramanian[8,10] proposed basing the prequantization step on an activity measure defined from the spatial arrangement of colors (Section 9.5). However, in this case, the computational cost of the prequantization step is no longer negligible, and the additional computation time induced by this step has to be compared to the computation time of the quantization algorithm without prequantization.

A last approach proposed by Kurz[42] estimates the color distribution of an image and its relative importance by examining the pixels at a fixed number of random locations. Kurz proposes the use of 1024 random locations for a 512×512 image. Randomizing the pixel locations avoids repeated selection of certain colors in the color set, particularly in images with periodic patterns.

9.3.2 Histogram calculation

The first preprocessing step of an image quantization algorithm generally consists of histogram computation of the image color set, i.e., a function H is computed such that $H(R, G, B)$ is equal to the number of pixels of the image whose color is equal to (R, G, B). Using the color space, one straightforward implementation of this function is to allocate a 256×256 array. Given this array, the computation of the histogram is performed by incrementing the corresponding entry of each pixel's color. However, this method has several disadvantages:

- First, the size of the image being quantized is generally much smaller than the size of the full histogram indicated above, and many entries of the histogram are set to 0 or very small values. This last property may induce some unwanted behavior for some algorithms. For example, determination of the most commonly occurring color may "lose" important colors if they are encoded in the histogram by a set of adjacent entries with low frequencies.
- Second, this encoding requires the storage of 256^3 indexes. Encoding each entry of the array by 2 bytes leads to a storage requirement of 32 MB.
- Finally, for many quantization algorithms using color spaces other than RGB, the allocation of an array enclosing the color gamut of these color spaces reinforces the problems linked with the sparse property of the histogram and its storage requirements.

One partial solution to this problem, proposed by Heckbert,[35] consists of removing the three least significant bits of each component so as to store the histogram in an array of size $32 \times 32 \times 32$. This solution reduces both the

memory requirements and the problems induced by the sparseness of the histogram. However, the removal of the three least significant bits is equivalent to prequantization, which may induce a visible loss of image quality when the output image is quantized into large number of colors such as 256 (Section 9.3.1). Moreover, using a color space other than RGB, one has to define the box enclosing the reduced color space. According to the color space, this box may remain large relative to the uniform quantization step.

One simple encoding of the histogram allocates an array whose size is equal to the number of different colors contained in the image. Each entry of this array is defined as a couple of fields encoding, respectively, a color contained in the image and the number of image's pixels of the corresponding color. This array may be initialized using a hash table and is widely used by tree-structured vector quantization algorithms (Section 9.4.2). Note that this last data structure is not designed to facilitate the determination of the number of image's pixels whose color is equal to a given value but rather to traverse the color space or a part of it to compute some of its statistical parameters such as the mean or the variance.

Xiang[77] proposes an encoding of the histogram based on a two-dimensional array of lists indexed by the R and G color coordinates. During the traversal of the image, each pixel with color (R, G, B) induces an update of the list of B values stored in the entry (R, G) of the array. If the component B is not present in the list, a new node is inserted with a color component storing the B value and a frequency field initialized to 1. If B is already present in the list, the frequency of the corresponding node is incremented. Each list is doubly linked and sorted in ascending order according to the blue field (see Figure 9.3a). Therefore, if S denotes the mean size of the blue lists, the retrieval of the number of pixels whose color is equal to a given (R, G, B) triplet requires $O(S/2)$ comparisons. Balasubramanian[10] improves the search by storing, in each entry of the (R, G) array, a binary tree ordered according to the blue values (see Figure 9.3b). The computational time required to retrieve a given (R,G,B) is then reduced to $O(\log(S))$.

9.4 *Clustering methods*

Clustering techniques perform a partition of the color space according to some criteria. Such criteria do not attempt to optimize the spatial

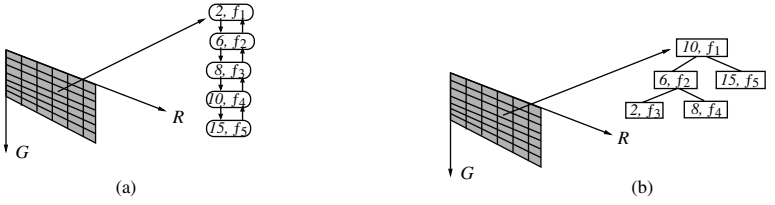


Figure 9.3 The histograms of (a) Xiang[77] and (b) Balasubramanian.[10]

arrangement of representatives, which may be improved by a dithering algorithm applied as a correcting step (Section 9.8). The input data of such algorithms are thus the image color set C_I and the frequency of occurrence of each color $\mathbf{c}$ in the image $f(\mathbf{c})$.

As anticipated in Section 9.1, quantization algorithms may be decomposed in two main families: post-clustering methods, which define a set of K representatives improved by an iterative scheme, and preclustering methods, which define a partition of C_I into K clusters and associate one representative to each cluster. Clustering quantization methods belong to the latter family. Each cluster $(C_i)_{i \in \{1, \ldots, K\}}$ of the partition may be characterized by statistical properties such as its mean, its variance along one coordinate axis, or its covariance matrix. All these parameters may be deduced from the following quantities:

$$
\begin{aligned}
card(C) &= \sum_{\mathbf{c} \in C} f(\mathbf{c}) \\
M_1(C) &= \sum_{\mathbf{c} \in C} f(\mathbf{c})\mathbf{c} \\
M_2(C) &= \sum_{\mathbf{c} \in C} f(\mathbf{c})(\mathbf{c}_1^2, \mathbf{c}_2^2, \mathbf{c}_3^2) \\
R_2(C) &= \sum_{\mathbf{c} \in C} f(\mathbf{c})\mathbf{c} \cdot \mathbf{c}^t
\end{aligned}
\tag{9.1}
$$

where $(\mathbf{c}_i^2)_{i \in \{1,2,3\}}$ denotes the squared value of the ith coordinate of the vector $\mathbf{c}$ and $f(\mathbf{c})$ denotes the number of pixels of the original image whose color is $\mathbf{c}$.

The quantities $card$(C), M_1(C) and M_2(C) are, respectively, called the cardinal, the first, and the second cumulative moments of the cluster. Note that $card$(C) is a real number, while M_1(C) and M_2(C) are three-dimensional vectors. The quantity $R_2(C)$ is a 3×3 matrix whose diagonal is equal to M_2(C).

The main advantage of the above quantities is that they can be efficiently updated during the merge or split operations performed by clustering quantization algorithms. For example, if two clusters, C_1 and C_2, must be merged, the cardinal of the merged cluster is equal to $card(C_1 \cup C_2) = card(C_1) + card(C_2)$. The same relation holds for M_1, M_2, and R_2. Conversely, if one cluster C is split into two sub-clusters C_1 and C_2, and if both the statistics of C and C_1 are known, the statistics of C_2 may be deduced from the ones of C and C_1. For example, the cardinal of C_2 is defined as $card(C) - card(C_1)$. The mean, the variances, and the covariance matrix of one cluster may be deduced from the cardinal, the moments, and the matrix R_2 by the following formula:

$$\mu = \frac{M_1(C)}{card(C)}$$

$$var_i = \frac{M_2(C)_i}{card(C)} - \mu_i^2 \qquad \forall i \in \{1, 2, 3\}$$

$$Cov = \frac{R_2(C)}{card(C)} - \mu \cdot \mu^t \tag{9.2}$$

where var_i and μ_i denote, respectively, the variance of cluster C along the coordinate axis Ω_i and the ith coordinate of the mean vector μ. The symbol Cov denotes the covariance matrix of the cluster.

Within the clustering quantization scheme, the covariance matrix of one cluster is used to determine the direction in which it spreads the widest. This direction, named the *major axis* of the cluster, is defined as the first eigenvector of the covariance matrix. Note that — the covariance matrix being real, symmetric, and positive — it can be diagonalized on an orthogonal basis. Each eigenvalue of the covariance matrix is equal to the variance of the cluster along the associated eigenvector (Figure 9.4). Moreover, the information held by one eigenvector is measured by

$$\frac{\lambda_i}{\sum_{i=1}^{3} \lambda_i} \tag{9.3}$$

where λ_i denotes the eigenvalue corresponding to eigenvector e_i.

(a) (b)

Figure 9.4 ***(See color insert following page 430)*** (a) Lenna test image and (b) its color set with the three eigenvectors (v_1, v_2, v_3) of its covariance matrix. The length of each vector is proportional to its eigenvalue.

Given a partition of C_I into K clusters $(C_i)_{i \in \{1, \ldots K\}}$, clustering quantization algorithms associate one representative $\mathbf{c}_i$ with each cluster C_i. The sum of quantization errors committed when mapping pixels whose colors fall in C_i to $\mathbf{c}_i$ is equal to the weighed sum of the squared distance between each color $\mathbf{c}$ in C_i and $\mathbf{c}_i$,

$$\sum_{\mathbf{c} \in C_i} f(\mathbf{c}) \|\mathbf{c} - \mathbf{c}_i\|^2$$

A well known statistical result[30] states that this sum is minimal when $\mathbf{c}_i$ is equal to the mean μ_i of C_i. The resulting sum of distances is called the *squared error* of C_i and may be understood as the error committed when assimilating C_i to its mean

$$\boldsymbol{SE}(C_i) = \sum_{\mathbf{c} \in C_i} f(\mathbf{c}) \|\mathbf{c} - \mu_i\|^2 \tag{9.4}$$

The squared error of one cluster C is related to the variances var_i computed along each coordinate axis $(\boldsymbol{\Omega}_i)_{i \in \{1, 2, 3\}}$ by the following formula:

$$\boldsymbol{SE}(C) = card(C) \sum_{i=1}^{3} var_i \tag{9.5}$$

Each quantity $(card(C) var_i)_{i \in \{1, 2, 3\}}$ represents the contribution of the axis i to the squared error of C and is called its *marginal squared error* along axis i. Note that, if the color space is defined from the three eigenvectors of the covariance matrix, the variance along each axis is equal to the eigenvalue of the associated eigenvector. We thus obtain

$$\boldsymbol{SE}(C) = card(C) \sum_{i=1}^{3} \lambda_i \tag{9.6}$$

where $(\lambda_i)_{i \in \{1, 2, 3\}}$ denotes the eigenvalues of the covariance matrix for the cluster.

The squared error of one cluster may be efficiently computed from its cardinal and the cumulative moments by the following formula (see Equations 9.2 and 9.5):

$$\boldsymbol{SE}(C_i) = \sum_{j=1}^{3} M_2(C_i)_j - \frac{M_1(C_i)_j^2}{card(C_i)} \tag{9.7}$$

where $M_1(C)_j$ and $M_2(C)_j$ denote, respectively, the jth coordinate of vectors $M_1(C_i)$ and $M_2(C_i)$.

Using the squared error, the total quantization error induced by a partition $\mathcal{P} = \{C_1, \ldots, C_K\}$ of the image color set is measured by the *total squared error* (TSE) defined as the sum of the squared errors of the clusters defining the partition,

$$E(\mathcal{P}) = \sum_{j=1}^{K} \boldsymbol{SE}(C_i) \tag{9.8}$$

The TSE criterion is usually used in three different ways:

1. Many papers on quantization use this criteria only *a posteriori* to demonstrate the efficiency of the proposed method.
2. This criterion is also used by some methods to improve *a posteriori* the color palette initially designed.
3. Finally, some authors base their quantization method on a minimization of the TSE. This category generally provides quantized images with a higher visual quality than the two previous ones. However, these algorithms are generally computationally intensive.

The different heuristics used to cluster colors may be decomposed into four main families:

1. The 3×1-D splitting methods described in Section 9.4.1 use the optimal algorithms defined for one-dimensional data to perform scalar quantization along each coordinate axis.
2. The 3-dimensional splitting methods split the three-dimensional image color set into a set of K clusters (Section 9.4.2). A large majority of these methods split recursively the initial image color set by using a top–down scheme.
3. Grouping methods, described in Section 9.4.3, use a bottom–up scheme by defining a set of empty clusters and aggregating each color of the image to one cluster.
4. Finally, merge methods described in Section 9.4.4 use a mixed approach by first splitting the image color set into a set of clusters and then merging these clusters to obtain the K required clusters.

9.4.1 3×1-D quantization methods

Using only one-dimensional data, a partition minimizing the total squared error may be computed with a complexity KM,[72,73] where M is the cardinality of the image color set. One first approach to take advantage of these optimal algorithms within the color quantization scheme computes the three marginal histograms of the image color set along each of the coordinate axis. The value $h_j(r)$ of the jth marginal histogram is defined as the number of pixels whose jth color coordinate is equal to r. Using the three marginal histograms, a scalar quantization algorithm is applied independently on each

axis of the color space. Such algorithms are referenced as *independent scalar quantization (ISQ) algorithms*.[38] However, because the ISQ uses only the marginal distribution of each scalar component, it cannot take inter-data correlation into account. As a result, many representatives are wasted in regions where the input colors have zero probability of occurrence, as shown in Figure 9.5a on a two-dimensional example.

Another approach, proposed by Balasubramanian[19] and named *sequential scalar quantization* (SSQ), consists of first performing a scalar quantization of the first coordinate axis Ω_1 into a predetermined number of levels K_1. This quantization of the first axis induces a partition $\mathcal{P}_1 = \{C_{21}, \ldots, C_{2K_1}\}$ of the image color set by planes orthogonal to Ω_1 as illustrated in Figure 9.5b. Then the marginal histograms along Ω_2 are computed for each cluster $(C_{2i})_{i \in \{1, \ldots, K_1\}}$. A scalar quantization based on these marginal histograms is performed and splits each cluster C_{2i} into n_{2j} sub-clusters separated by planes orthogonal to Ω_2. This quantization step produces a total of K_2 clusters $\mathcal{P}_2 = \{C_{3i}, \ldots, C_{3K_2}\}$. Finally, a last scalar quantization along Ω_3 is performed on the third marginal histogram of each cluster $(C_{3i})_{i \in \{1, \ldots, K_1\}}$, splitting it into n_{3j} sub-clusters. This last quantization step produces the required number K of clusters.

As claimed by the authors, this method has a lower computational cost than most existing quantization methods. However, it raises some problems only partially solved by the authors. First, the final total squared error is dependent on the order in which the scalar quantizations are performed (we have tacitly assumed that the Ω_j are quantized in the order Ω_1, Ω_2, Ω_3). The only means of finding the best order is to apply the quantization scheme on each of the 3! = 6 possible orders. The authors address this problem by using the YCrCb color space with one luminance (Y) and two chrominance (C_r and C_b) axes. The authors first perform two scalar quantizations on the chrominance plane (C_r, C_b) followed by one scalar quantization of the luminance axis Y. This strategy being based on a fixed order of the quantizations may lead to suboptimal results. The second problem raised by this method is the determination of the number of quantization levels along each axis. In other words, what values of K_1, K_2, $(n_{2_j})_{j \in 1, \ldots, K_1}$, and $(n_{3_j})_{j \in 1, \ldots, K_2}$ should be picked

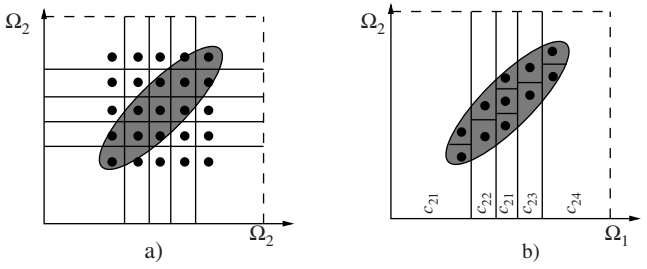


Figure 9.5 (a) Independent scalar quantization into $K = 25$ levels and (b) sequential scalar quantization into $K = 11$ levels.

to obtain the required number K of final clusters? The authors estimate these quantities by using results from the asymptotic quantization theory. This theory being valid only for very large values of K, the authors perform a preliminary quantization for some initial choice of K_1 and K_2 and correct these initial values as follows:

$$\left.\begin{aligned} K_1 &= K_1^0\left(\frac{d_1^2}{d_2 d_3}\right)^{\frac{1}{6}} \\ K_2 &= K_2^0\left(\frac{d_1 d_2}{d_3}\right)^{\frac{1}{6}} \end{aligned}\right\} \text{with } \forall j \in \{1,2,3\} \quad d_j = \sum_{i=1}^{K} \sum_{\mathbf{c} \in C_i} f(\mathbf{c})(\mathbf{c}^j - \mathbf{c}_i^j)^2 \qquad (9.9)$$

where K_1^0 and K_2^0 are the initial choices for K_1 and K_2, and d_1, d_2, and d_3 denote the marginal total squared errors along each axis. Symbols $(\mathbf{c}^j)_{j \in \{1,2,3\}}$ and $(\mathbf{c}_i^j)_{j \in \{1,2,3\}}$ denote, respectively, the jth coordinate of the color vector $\mathbf{c}$ and the ith representative $\mathbf{c}_i$.

9.4.2 *Three-dimensional splitting methods*

Various three-dimensional splitting methods have been intensively explored[8,9,12–13,14,70–72,74,75] since 1982, and the median cut method was proposed by Heckbert.[35] These methods create a partition of the image color set into K clusters by a sequence of $K - 1$ split operations.

The initial image color set is thus split by a set of planes named *cutting planes*, each plane being normal to one direction named the *cutting axis*. The location of the cutting plane along the cutting axis is named the *cutting position*. One justification for the use of planes within the three-dimensional splitting scheme is provided by the inverse colormap operation. Indeed, as mentioned in Section 9.4, given a set of representative colors $\{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$, an optimal mapping with respect to the set of representatives maps each initial color to its closest representative. This optimal mapping induces a partition of the image color set by a three-dimensional Voronoï diagram[60] defined by the representatives. Each cell of a three-dimensional Voronoï diagram being delimited by a set of planes, the use of planes within the three-dimensional splitting scheme does not induce a loss of generality. Note that, if each plane is assumed to be perpendicular to one of the coordinate axes, each cluster is a hyperbox.

As mentioned in Section 9.1, the set of all possible partitions into K clusters of the initial set of M data points is too large for an exhaustive enumeration. The three-dimensional splitting scheme must thus use a set of heuristics to restrict the set of possible partitions. The main heuristics used by splitting quantization algorithms may be decomposed into four steps common to all algorithms of this family:

1. Selection of a splitting strategy
2. Selection of the next cluster to be split

3. Selection of the cutting axis
4. Selection of the cutting position

The remainder of this section describes the main heuristics used for each of the above steps. The different choices performed by the methods described in this section are summarized in Table 9.1.

Table 9.1 Different Strategies Used by Five Tree-Structured Vector Quantizers

		1	2	3	4	5
Cluster selection	Greatest squared error		*		*	*
	Greatest eigenvalue			*		
	Greatest cardinality	*				
Cutting axis	Longest coordinate axis	*				
	Coordinate axis with greatest variance					*
	Major axis		*	*	*	
Cutting position	Median cut	*				
	Marginal squared error minimization		*			
	TSE minimization				*	*
	Pass through the mean			*		

Note: 1 = Heckbert,[35] 2 = Wan and Wong,[71] 3 = Bouman and Orchard,[12,13] 4 = Wu,[75] and 5 = Braquelaire and Brun.[14]

9.4.2.1 *Splitting strategy*

A large majority of 3-D splitting methods[8,12,13,70–72,75] recursively split the initial image color set into two sub-clusters until the K final clusters are obtained. This bipartitioning strategy may be encoded by a complete binary tree, and quantizers following this scheme are called *tree-structured vector quantizers*.[74] The internal nodes of this tree encode intermediate clusters and have exactly two siblings. The final clusters are encoded by the leaves of the binary tree (Figure 9.6).

The number of recursive splits that may be performed using the bipartitioning strategy is equal to the number of binary trees having exactly K leaves,[28]

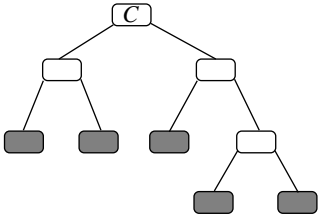


Figure 9.6 A complete binary tree defining a partition into five clusters.

$$\frac{1}{K}\binom{2(K-1)}{K-1}$$

This number is typically too large for an exhaustive enumeration. Chou et al.[21] and Lin et al.[44] create a binary tree of N leaves with

$$K < N << \frac{1}{K}\binom{2(K-1)}{K-1}$$

and then prune the tree so as to select the K leaves that induce the lowest partition error. However, according to Wu,[74] this strategy incurs a high overhead as compared with a strategy generating only the K required clusters and does not induce a significant decrease of the total squared error.

The main drawback of the bipartitioning scheme is that each bipartitioning is performed regardless of its impact on further subdivisions performed deeper in the binary tree. This greedy local criterion may contradict the total squared error criterion, which should be globally minimized. Wu[74] proposed performing a first splitting step of the image color set by κ planes normal to the major axis of the color set. The relative position of these planes along the major axis are globally optimized using a dynamic programming scheme. The $\kappa + 1$ generated clusters are then recursively split into two sub-clusters using the recursive bipartitioning scheme described previously until the final number K of clusters is generated. The value of the critical parameter κ is estimated during the construction of the $\kappa + 1$ clusters. According to experiments performed by Wu, this value falls between four and eight, according to the distribution of the image color set.

9.4.2.2 Cluster selection

Using a bipartitioning scheme, each iteration of the quantization algorithm selects one leaf of the binary tree and splits it into two sub-clusters. The criterion used to select the cluster to be split varies according to the criteria minimized by the quantization algorithm.

The median cut algorithm proposed by Heckbert[35] divides the color space Ω into rectangular clusters C_k with the same cardinality. Therefore, to equalize the cardinality of clusters, Heckbert selects at each step the cluster with the greatest cardinality. Several splitting algorithms based on $k-d$ trees[11] (Section 9.7.3) use an approach similar to the one of Heckbert. These algorithms assign an approximately equal number of colors to all leaves of the tree. However, such a counterbalancing is not really adapted to image quantization. Indeed, there is no relevant justification to require that each cluster should contain a nearly equal number of colors while ignoring how these colors are distributed in the color space.[70] Using such a method, a cluster with a large quantization error may not be split, while a cluster containing only one color (with a high occurrence frequency) may be subdivided instead.

A large majority[12–14,70–72,74,75] of methods attempt to minimize the total squared error. To obtain homogeneous clusters, Bouman[13] selects, at each iteration, the cluster whose data spread the widest along the splitting direction. Because Bouman splits clusters along their major axis, the method selects the cluster whose principal eigenvalue is maximal.

Wan et al.[70] proposed splitting, at each step, the cluster with the greatest squared error. The basic idea of this strategy is to split the cluster whose contribution to the TSE is the largest. This strategy may be compared to the one of Bouman by writing the squared error into the base defined by the three eigenvectors of the covariance matrix,

$$\boldsymbol{SE}(C) = card(C)\sum_{i=1}^{3}\lambda_i$$

The heuristic of Bouman may thus be understood as an approximation of the one of Wan, neglecting the possible decrease of the variance along the directions defined by the two remaining eigenvectors.

One slightly different strategy has been proposed by Wu.[75] It consists of selecting, at each iteration, the cluster whose bipartitioning yields the largest reduction of the total squared error. Given a partition of the image color set into k clusters $\mathcal{P}_k = \{C_1, \ldots, C_k\}$, the splitting of one cluster C_i into two subclusters C_1^1, and C_i^2 modifies the total squared error as follows:

$$E(\mathcal{P}_{k+1}) = E(\mathcal{P}_k) + \boldsymbol{SE}(C_i^1) + \boldsymbol{SE}(C_i^2) - \boldsymbol{SE}(C_i) \tag{9.10}$$

Note that, if both C_i^1 and C_i^2 are non-empty, the value $\boldsymbol{SE}(C_i^1) + \boldsymbol{SE}(C_i^2) - \boldsymbol{SE}(C_i)$ is strictly negative. Therefore, the total squared error is a strictly decreasing function of the number of split operations.

Wan's heuristic may thus be considered as an approximation of the heuristic of Wu, neglecting the squared errors of the two generated clusters. However, using the heuristic of Wu, each cluster must be split to estimate the decrease of the total squared error. According to experiments performed by Wu,[74] the difference between the two criteria in terms of final quantization errors is marginal, while the increase in computational costs is twofold for the second criteria. The selection at each iteration of the cluster with the greatest squared error is thus a good compromise between the computational time and the final quantization error.

9.4.2.3 Cutting axis

Given the cluster (C_i) to be split, one has to determine the normal and the location of the cutting plane. Because the decrease of the total squared error after the split of (C_i) into C_i^1 and C_i^2 is equal to $\boldsymbol{SE}(C_i^1) + \boldsymbol{SE}(C_i^2) - \boldsymbol{SE}(C_i)$, the optimal cutting plane is the one that minimizes $\boldsymbol{SE}(C_i^1) + \boldsymbol{SE}(C_i^2)$. However, the freedom to place such a plane is still too enormous, and we need

to fix the orientation of this plane to make the search feasible. Because the cutting of the cluster mainly decreases the variance along the cutting axis, one common heuristic consists of selecting the cutting axis among the directions with a high variance (Equation 9.5).

Heckbert[35] approximates the variance by enclosing the cluster in a rectangular box. The two sides of this box normal to the coordinate axis Ω_i enclose the color vectors with minimum (resp. maximum) coordinates along Ω_i. The spread of the data along each coordinate axis is then estimated by the length of the box along this axis, and the box is cut perpendicularly to its longest axis.

Braquelaire et al.[14] compute the variance along each axis and split each cluster along its coordinate axis with the greatest variance. This heuristic decreases the variance along the coordinate axis, which brings the major contribution to the squared error. One computational advantage of this heuristic is that, with the dataset being split along three fixed directions (the coordinate axis), the image color set may be preprocessed to optimize the determination of the cutting plane.[14] However, the coordinate axis with the greater variance is generally not the axis along which the dataset spreads the widest. This direction is provided by the major axis of the cluster and the relation between the squared error of the cluster, and the three eigenvalues of the covariance matrix is provided by Equation 9.6. Note that if λ_1 denotes the principal eigenvalue of the major axis, we have

$$(\forall i \in \{1, 2, 3\}) \qquad \lambda_1 \geq var_i$$

Therefore, a greater decrease of the total squared error may be expected by cutting the cluster along its major axis. This last heuristic is used by Wan,[70,71] Wu,[74,75] and Bouman.[12,13]

9.4.2.4 *Cutting position*

Given a selected cluster C and one cutting axis defined by a unit vector A, let us denote by m and M, respecitvely, the minimal and maximal projections of C on the cutting axis. Because the cutting plane is normal to A, its projection on the cutting axis is equal to one real number t. Any value of $t \in [m, M]$ causes a bipartition of C into two sub-clusters C_t^1 and C_t^2 as illustrated in Figure 9.7. Note that, if we assume that the cutting plane belongs to C_t^2, we have $card(C_m^1) = 0$ and $card(C_m^2) = card(C)$ while $card(C_{M+\varepsilon}^1) = card(C)$ and $card(C_{M+\varepsilon}^2) = 0$ where ε is any positive real number. More precisely, the function δ defined by

$$\delta(t) = \frac{card(C_t^1)}{card(C)}$$

is an increasing function of t from $[m, M]$ to $[0, 1]$.

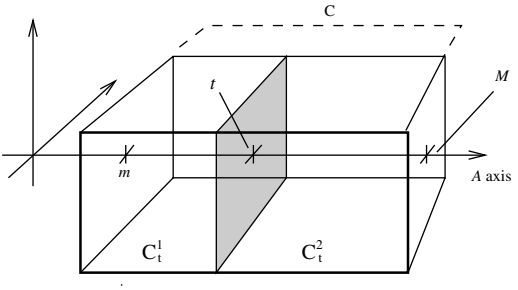


Figure 9.7 Splitting of one cluster along axis A.

The value of t such that $\delta(t) = 1/2$ corresponds to a split of the initial cluster into two sub-clusters with the same cardinality. The median cut algorithm of Heckbert[35] defines a partition into K clusters with an equal cardinality. Therefore, given the selected cluster and the cutting axis, Heckbert sweeps the cutting plane along the cutting axis from $t = m$ until $\delta(t) = 1/2$.

Bouman[12,13] selects as the value of t the projection of the mean on the cutting axis: $\mu \cdot A$, where μ is the mean of C. This strategy avoids the sweeping of the cutting plane along the cutting axis. However, the validity of this choice is demonstrated only for large clusters with Gaussian distribution.

Wan et al.[70] maximize the decrease of the marginal squared error along the major axis (Section 9.4 and Equation 9.5). The optimal value t_{opt} maximizing the decrease of the marginal squared error along the major axis is defined by[70]

$$t_{opt} = \arg \max_{t \in [m, M]} [var - (\delta(t) var_t^1 + (1 - \delta(t)) var_t^2)]$$

where var_t^1 and var_t^2 denote, respectively, the variance of C_t^1 and C_t^2 along the major axis of C. The above formula uses both parameters from clusters C_t^1 and C_t^2. This last property induces an update of the parameters of C_t^1 and C_t^2 when sweeping the cutting plane from $t = m$ to $t = M$. Wan et al.[70,72] proved that the cutting position is also provided by

$$t_{opt} = \arg \max_{t \in [m, M]} \left[\frac{\delta t}{1 - \delta(t)} ((\mu - \mu_t^1) \cdot A)^2 \right] \tag{9.11}$$

where μ and μ_t^1 denote, respectively, the mean of C and C_t^1.

The main advantage of this last formula is that t_{opt} is now only a function of the parameters of C_t^1 that may be incrementally updated when sweeping the plane along the major axis of C. Moreover, using theoretical results from scalar quantization,[72] Wan et al. show that the interval $[m, M]$ may be restricted to

$$\left[\frac{m+\mu\cdot A}{2}, \frac{M+\mu\cdot A}{2}\right]$$

Note that this last interval encloses the cutting position $\mu \cdot A$ selected by Bouman. Nevertheless, Wan et al. maximize only the decrease of the marginal squared error along the major axis of the cluster while, according to Equation 9.6, the squared error is equal to the sum of the marginal squared errors along the three eigenvectors.

Wu[74,75] maximizes the decrease of the total squared error (Equation 9.10) defined by $\boldsymbol{SE}(C_t^1) + \boldsymbol{SE}(C_t^2) - \boldsymbol{SE}(C)$. Wu proved that this last formula is minimal for a value of t_{opt} defined by

$$t_{opt} = \arg \max_{t \in [m, M]} \frac{\left\|M_1(C_t^1)\right\|^2}{card(C_t^1)} + \frac{\left\|M_1(C) - M_1(C_t^1)\right\|^2}{card(C) - card(C_t^1)} \tag{9.12}$$

This last expression uses only the cardinality and the first moment of cluster C_t^1 that can be efficiently updated when sweeping the cutting plane. However, the uses of the squared value of the first moment may lead to rounding errors. Brun[14,16] has shown that the value t_{opt} maximizing the decrease of the partition error may also be defined as

$$t_{opt} = \arg \max_{t \in [m, M]} g(t) \text{ with } g(t) = \frac{\delta(t)}{1-\delta(t)}(\mu - \mu_t^1)^2 \tag{9.13}$$

This new formulation may be considered as an extension of Equation 9.11 to the three-dimensional case. It allows us to manipulate smaller quantities than Equation 9.12 and thus avoids rounding errors induced by this last formula.

Brun has also shown that the function $g(t)$ is bounded above by one parabola U and below by a hyperbola L (see Figure 9.8). The parabola U allows us to stop the search of the optimal value t_{opt} as soon as $\delta(t) \geq \delta_{max}$ as indicated in Figure 9.8. According to experiments performed by Brun, the mean reduction of the interval $[m, M]$ with a quantization into 16 colors is about 10%. Note that, becausee the functions U and L reach their maximum when δ is equal to $1/2$, the maximum of $g(t)$ is reached for a value of t such that $\delta(t)$ is about $1/2$. The value $\delta = 1/2$ corresponds to the median position of the splitting plane selected by Heckbert's median cut algorithm.

9.4.3 *Grouping methods*

Grouping techniques generate the K representative colors during one traversal of the image. Using such a scheme, each representative is defined according to the representatives already selected and the current color set. Indeed, the color set used at each iteration of the algorithm may be restricted to the already scanned colors. Alternative techniques[31] define K representa-

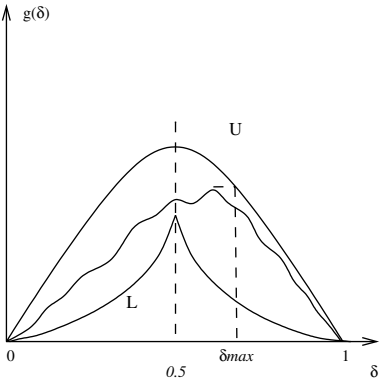


Figure 9.8 Evolution of the total squared error induced by the split of one cluster by a plane as a function of the cardinality of one of the two clusters.

tives by scanning the K first colors of the image and updating this set during the traversal of the image.

9.4.3.1 *The merge and box algorithm*

The merge and box algorithm, proposed by Fletcher,[25] is based on an iterative process that is repeated until the boxes with the K representative colors are formed. This algorithm iterates the following steps: each distinct color of the color space is placed into a sub-box of side length equal to one. If the input color can be approximated with an error no greater than half the length of box's diagonal, by one of the sub-boxes already generated, this color is included in the corresponding sub-box. Otherwise, a new sub-box of side one is added to the list. To obtain a list that does not exceed K sub-boxes, a merging process is used when the number of sub-boxes is equal to $K + 1$ to gather the pair of sub-boxes, which must be merged into a single sub-box. The merging pair is the one that minimizes the largest side of the merged sub-box.

According to Fletcher,[25] the heuristic used by the merger generates roughly cubical sub-boxes with a similar size. Using a Manhattan distance between corners would result in elongated sub-boxes, causing larger quantization errors. This algorithm may be parallelized on a SIMD parallel processor. Moreover, its calculating time may be further reduced by using a spatial sampling to approximate the image color set accurately.

Gervautz[31] proposed a similar approach based on a hierarchical decomposition of the RGB cube by an octree. Using this decomposition, each cube of side one used by Fetcher is encoded as a leaf of the octree. Gervautz builds the first K leaves of the octree by scanning the first K colors of the image. Each leaf of the tree thus represents one cluster of the RGB color space. If an additional color belongs to one of these clusters, the mean color of the associated leaf is updated, and the structure of the octree remains unchanged.

Otherwise, a new leaf is created from the merger of its children so as to keep only K leaves. The set of K representative colors is then defined from the centroid of the clusters associated with each leaf of the octree.

9.4.3.2 *The max-min algorithm*

The max-min algorithm, first proposed by Houle and Dubois,[37] iterates the following steps (see Figure 9.9):

1. A single color c_1 is first selected from the original image. It could be the most frequently occurring one.
2. A new representative color c_k is then computed; c_k is the unselected color whose minimum distance to any of the representative colors thus far selected is maximal; i.e., c_k must verify

$$\min_{k' = 1, \ldots, k-1} \|c_k - c_{k'}\|^2 \geq \min_{k' = 1, \ldots, k-1} \|c - c_{k'}\|^2 \quad \forall \mathbf{c} \in \{\Omega^I - \Omega^Q\}$$

where symbols Ω^I and Ω^Q denote, respectively, the image color set and the set of representatives already selected.

Step 2 of the max-min algorithm is iterated until K representative colors are selected.

A similar approach has been proposed by Xiang,[76] who initializes the first color c_1 arbitrary. Note that an arbitrary selection of the first color avoids computation of the image's histogram. This quantization scheme has also been investigated by Trémeau et al.[64] to extend it to a vector quantization process based on the nearest color query principle.

According to Houle and Dubois, the set of representatives generated by this algorithm is uniformly distributed and extends right to the boundary of the image color set. This distribution is well suited when a pseudo-random noise vector is added to the color value of the pixel to be quantized, i.e.,

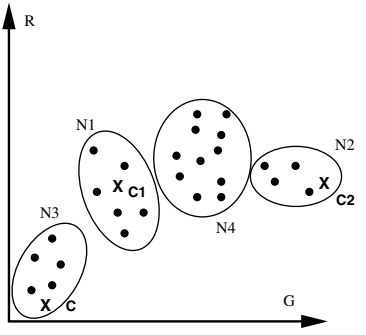


Figure 9.9 The max-min algorithm. In this example, the algorithm first selects the most commonly occurring color, c_1, and then c_2 and c according to the distance to the previously selected colors.

when we try to minimize errors of quantization by dithering the quantized image (Section 9.8). In this last case, the max-min algorithm gives better depiction of low contrast details — but at the expense of granularity due to the pseudo-random quantization.

9.4.4 *Merge methods*

Methods described in Section 9.4.2 split recursively the image color set C_I until the required number K of clusters is obtained. Such methods obtain generally quantized images with high visual quality but must traverse each color $\log_2(K)$ times. On the other hand, the grouping methods described in Section 9.4.3 traverse each pixel of the original image to map its color to one cluster. Therefore, such methods traverse the original image only once. However, the function used to map each color to one cluster must have a very low computational cost to achieve overall low processing times. This constraint forbids the use of complex heuristics in the mapping function.

Merge methods may be understood as intermediate methods between pure split or pure grouping methods. Such methods perform a first splitting step so as to define a set of $N > K$ clusters and then perform a sequence of merges to obtain the K final clusters. The first splitting step has to reduce the number of data points while keeping the main properties of the image color set. Based on this reduced set of data, the merge algorithm may apply heuristics with a higher computational complexity than the ones designed within the grouping quantization framework. The set of clusters, along with their adjacency relationships, may be thought of as a graph where each vertex encodes one cluster, and each edge an adjacency relation between two clusters. Then, the merge of two clusters is equivalent to the contraction of the edge encoding their adjacency and the removal of any multiple edges between adjacent clusters. The value of each edge encodes the distance between two clusters. After the merge of two clusters, the value of the edges incident to the newly created cluster has to be updated. Using the TSE (Equation 9.8), the value of each edge is defined as the increases of the total squared error induced by the merge of the two adjacent clusters.

If $\mathcal{P}$ and $\mathcal{P}'$ denote, respectively, the partition of the image color set before and after the merge of two clusters C_k and $C_{k'}$, we have

$$\mathcal{P}' = (\mathcal{P} - \{C_k, C_{k'}\}) \cup \{C_k \cup C_{k'}\}$$

and the total squared error of $\mathcal{P}'$, $E(\mathcal{P}')$, may be deduced from $E(\mathcal{P})$ by

$$E(\mathcal{P}') = E(\mathcal{P}) + \frac{card(C_k) \cdot card(C_{k'})}{card(C_k) + card(C_{k'})} \|\mu_k - \mu_{k'}\|^2 \tag{9.14}$$

The value of an edge encoding an adjacency relationship between clusters C_k and $C_{k'}$ is thus equal to

$$\Delta_{k,k'} = \frac{card(C_k) \cdot card(C_{k'})}{card(C_k) + card(C_{k'})} \|\mu_k - \mu_{k'}\|^2 \tag{9.15}$$

Note that one is effectively minimizing a weighted Euclidean distance between the two centroids μ_k and $\mu_{k'}$ of the two clusters C_k and $C_{k'}$. This is the reason why such methods are often referred to as pair-wise nearest neighbor clustering.[10,23]

The main computational cost of merge methods comes from the merger that has to traverse all pairs of adjacent clusters at each step. Two approaches have been proposed and sometimes combined[10] to reduce this computational cost: the reduction of the number of initial clusters and the use of heuristics in the merge strategy.

Brun and Mokhtari[17] first perform a uniform quantization of the RGB cube into N clusters and then create a complete graph from the set of clusters. This graph is reduced by merging, at each step, the two closest clusters according to Equation 9.14. Experiments performed by Brun et al. show that, if K is lower than 16, very low values of N, such as 200 or 300, are sufficient to obtain high-quality quantized images.

Dixit[22] reduces the initial number of clusters by a subsampling of the image at random locations. According to Dixit, as few as 1024 random locations on a 512×512 image are sufficient to obtain a good estimate of the distribution of the image color set. The set of clusters is then sorted in ascending order according to the cardinality of each cluster. The sorted table of clusters is then traversed from the clusters with lowest cardinality, and each current cluster is paired with the closest remaining one according to Equation 9.14. These two clusters are then excluded from the merge. Therefore, no cluster is allowed to be paired with more than one cluster, and the set of clusters is reduced by a factor two at each iteration. Note that this fixed decimation ratio between two successive iterations may lead to the merger of clusters that are far apart.

Xiang and Joy[77] create the initial set of clusters by using a uniform quantization of the RGB color space into N clusters. Xiang reduces the N initial clusters to K final clusters by a sequence of $N - K$ merges. At each step, the merge criterion selects two clusters such that the newly created cluster is enclosed by a minimal bounding box. Xiang thus estimates the homogeneity of one cluster by the size of its bounding box. Using an initial uniform quantization into $2 \times 1 \times 4$ boxes, each cluster has approximately the same luminance in the *YIQ*–NTSC system. Therefore, the size of a bounding box may be understood as the shift in luminance within the associated cluster, and the Xiang algorithm mainly attempts to minimize the shift in luminance within the quantized image. Such shifts may produce significant visual artifacts in the ray-traced images used by Xiang.

Equitz[23] reduces the computational cost of the merger by using a $k - d$ tree (see Section 9.7.3), which decomposes the set of initial clusters into a set of regions each composed of eight clusters. The complete graph defined by

Brun[17] is thus decomposed into a set of non-connected complete sub-graphs, each composed of eight vertices. Then, Equitz finds the two closest clusters in each sub-graph according to Equation 9.14 and merges a fixed fraction of them (such as 50%). After each iteration, the decomposition of the initial graph into regions is also updated so as to obtain roughly equal size sub-graphs. Note that this merge strategy neglects adjacencies between vertices belonging to different sub-graphs. Equitz's heuristic has been improved by Balasubramanian and Allebachm,[10] who first perform a prequantization step described in Section 9.5. Balasubramanian also weights the distance between two clusters by an activity measure defined from the initial image (Section 9.5).

9.4.5 *Popularity methods*

This family of methods, introduced by Heckbert,[35] uses the histogram of the image color set to define the set of representatives as the K most occurring colors in the image. This simple and efficient algorithm may, however, perform poorly on images with a wide range of colors. Moreover, it often neglects color in sparse regions of the color set. Likewise, this algorithm performs poorly when asked to quantize to a small number of colors (say, <50).[35]

This algorithm was modified and optimized by Braudaway[15] in 1986. Braudaway encodes the image histogram by a uniform partition of the color space Ω into $L \times L \times L$ sub-boxes Ω_k of equal size N/L (see Section 9.3.2 and Figure 9.10). To prevent the next representative color from being chosen too close to the previous one, a reduction function is applied to the histogram after the selection of each representative. The cardinal $card(\Omega_k)$ (Equation 9.1) of each sub-box Ω_k is then reduced by a factor $(1 - e^{-\alpha r^2})$ where r is the distance between Ω_k and the previously selected color.

The degree of histogram reduction for a fixed value of r is controlled by α. In the investigation done by Braudaway, α was chosen so that the histogram was reduced by a factor of $1/4$ at a distance $r = N/4$ between the studied sub-box Ω_k and the previous selected color.

9.5 *Quantization algorithms based on weighted errors*

We saw, in Section 9.4.2, several methods based on a minimization of the total squared error. The use of the total squared error within the quantization framework relies on the assumption that a partition of the image color space into homogeneous clusters produces an output image visually closed to the original. This assumption is partially justified by the following equation:

$$E(C) = \sum_{i-1}^{K} \boldsymbol{SE}(C_i) = \sum_{(i,j) \in \{1,\ldots,m\} \times \{1,\ldots,n\}} \|I(i,j) - I'(i,j)\|^2 \qquad (9.16)$$

where $m \times n$ denotes the size of the images and $I(i, j)$, $I'(i, j)$, denote, respectively, the color of the pixel (i, j) in the original and quantized images.

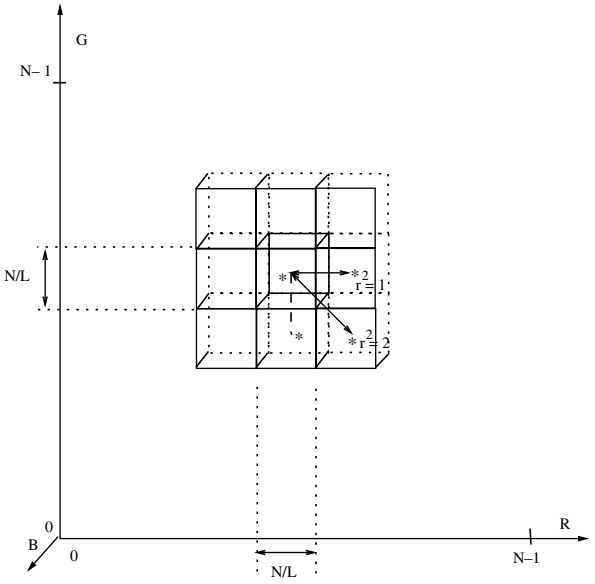


Figure 9.10 The reduction factor applied by Braudaway's algorithm.[15] The color space is enclosed in an $N \times N \times N$ box and subdivided into L^3 sub-boxes. The symbol r denotes the distance of each sub-box from the center one.

The total squared error may thus be understood as the sum of the squared distances between the pixels of the original image and the quantized one. Despite this interesting property, the total squared error should not be considered as a visual distance between the original and quantized images. This is confirmed by the experiments displayed in Figure 9.11. The 55,510 different colors of Figure 9.11a are reduced to 8 colors by a quantization algorithm.[17] The quantized image is displayed in Figure 9.11b. Figure 9.11c is obtained from 9.11b with an additional dithering step performed by the Flyod–Steinberg[27] algorithm (Section 9.8). Although Figure 9.11c seems visually closer to 9.11a than 9.11b, the total squared error of Figure 9.11c

(a) (b) (c)

Figure 9.11 ***(See color insert)*** (a) The original image, (b) the quantized image with eight colors, and (c) image improved with a dithering algorithm.

computed, per Equation 9.16, is nearly twice as large as the one of Figure 9.11b. This surprising result is due to the different criteria used by the dithering and quantization algorithms. The dithering algorithm attempts to optimize the local arrangement of representatives, while the quantization algorithm minimizes the global homogeneity of clusters.

One method to take into account the local properties of the image during the quantization step is to replace the frequency function f in the squared error's equation (Equation 9.4) by a weighted frequency function W, where $W(c)$ is defined as a sum of local attributes computed on each pixel with color c. Note that, from this point of view, the usual squared error may be understood as a special case of weighted squared error where the weight of each color is equal to the number of pixels mapped to it. Given an initial color space C partitioned into K clusters $\{C_1, \ldots, C_K\}$, the weighted TSE is thus defined by:

$$E(C) = \sum_{i=1}^{K} \sum_{c \in C_i} W(c) \|c - \mu_i\|^2 \tag{9.17}$$

One prominent artifact of a limited palette size is false contouring. False contours occur when a smoothly varying region is mapped into a small number of colors from the colormap. Instead of displaying slow variations of colors across the region, smaller regions with a constant color are displayed with abrupt changes across region boundaries. These abrupt color changes are perceived as contour in the quantized image. Bouman[12,13] estimates the size of uniformly quantized regions by using the variations of the luminance defined in the NTSC system,

$$y(p) = [0.300, 0.586, 0.115] I(p)^t \tag{9.18}$$

where the color of pixel p, $I(p)$, is defined in the RGB color space.

Bouman has shown that, within a region with a linear variation of the luminance with slope ∇y, the ratio between the distance of each pixel's color to its representative and the local gradient of the luminance is proportional to the width of the uniformly mapped regions. More precisely, we have

$$\sum_{p \in R_n} \frac{\|I(p) - c_n\|^2}{\|\nabla y(p)\|^2} \approx |R_n| \frac{\Delta^2}{12}$$

where R_n denotes the set of pixels p whose color $I(p)$ is mapped to c_n. The cardinality of this set is denoted by $|R_n|$. The symbol Δ denotes the width of a uniformly quantized region in the direction of the luminance gradient ∇y.

Therefore, to minimize the size of the uniformly mapped regions (estimated by Δ), Bouman uses a weighting inversely proportional to the

luminance gradient of the image. For each color c of the original image, the weighting $W(c)$ is defined by

$$W(c) = \sum_{p \in I | I(p) = c} \left(\frac{1}{h \times \min\{\|\nabla y(p)\|, 16\} + 2} \right)^2 \tag{9.19}$$

where the constants 16 and 2 bound the dynamic range of the gradient, while the function h is a smoothing kernel, the convolution of which allows us to estimate the mean gradient over the regions.

Note that any scale of the luminance component in the original image by a factor s scales the weighting factor $W(sc)$ by $1/s^2$, while the squared distance between c and its representative is scaled by a factor s^2. Therefore, the factors $W(c)\|c - c_i\|^2$ and the weighted TSE (see Equation 9.17) are insensitive to any global scaling of the luminance. This obeys Weber's law,[55] which suggests that any measure of subjective image quality should be invariant to absolute intensity scaling.

Bouman integrates the weighting defined by Equation 9.19 into a tree-structured vector quantizer (Section 9.4.2 and Table 9.1). Using the weighted TSE (Equation 9.17), the weighted moments of a cluster C are simply defined by substituting $f(c)$ by $W(c)$ in Equations 9.1. The weighted mean and covariance of the cluster are then defined from the weighted moments using Equation 9.2. Bouman uses the weighted mean μ_i^w and covariance matrix Cov_i^w of each cluster C_i to adapt his tree-structured vector quantizer as follows:

1. Select the cluster C_m whose weighted covariance matrix has the greatest eigenvalue. Let us denote by e_m the eigenvector associated with this eigenvalue.
2. Split the selected cluster by a plane that is orthogonal to e_m and passing through μ_m^w.

Note that, because the introduction of weights modifies the distribution of the initial image color set, it modifies all key steps of Bouman's recursive split algorithm. Moreover, the weights introduced by Bouman may be easily adapted to other recursive splitting schemes. For example, a weighted minimization of the TSE may be achieved by computing the weighted squared error using the weighted moments (see Equations 9.19 and 9.7) and selecting, at each step, the cluster whose weighted squared error is maximal.

One other characteristic of the human visual system is its greater sensitivity to quantization errors in smooth areas than in busy regions of the image. Balasubramanian and Allebach[8,10] define a color activity measure on each pixel by

$$\alpha_c = \frac{1}{64} \sum_{p \in P_k} \left\| c_p - \bar{c_k} \right\|$$

where P_k is an 8×8 block centered around pixel k. The symbols c_p and $\bar{c}_k$ denote, respectively, the color of pixel p and the mean color of P_k.

These values are then accumulated on colors by associating with each color the minimal color activity measure of the color over all spatial blocks,

$$\tilde{\alpha}_c = \min_{k \in K_c} \alpha_k \tag{9.20}$$

where K_c denotes the set of blocks in which color c occurs.

Because the luminance component has the greatest variations, the gradient of the luminance is also used as an alternative[9] or to complement[10] the measure of the spatial masking. Experiments show that the masking of quantization noise by image activity at a pixel depends not only on the luminance gradient at the pixel but also on gradients at the neighboring pixels. Based on these experiments, Balasubramanian[9,10] computes the gradient ∇y of the luminance defined in the NTSC system (Equation 9.18) on each pixel. Then, the luminance activity α_k^l of each 8×8 block k is defined as an average value of its gradient. The luminance activity of each color is then defined as the average value of the block's gradient in which this color occurs,

$$\tilde{\alpha}_c^l = \frac{1}{|K_c|} \sum_{k \in K_c} \alpha_k^l \tag{9.21}$$

where K_c denotes the set of blocks in which color c occurs.

The color or luminance activity may then be used to weight the colors during the quantization step. Balasubramanian chose

$$W(c) = \frac{1}{\tilde{\alpha}_c}$$

and

$$W(c) = \frac{1}{\tilde{\alpha}_c^2}$$

respectively, in References 9 and 10. In both cases, the importance of colors with low activity measures is reinforced by the weighting, to the detriment of colors with high activity measures whose quantization errors are less readily perceived by the human visual system.

Balasubramanian uses the color and luminance activity measures within his quantization algorithm based on a merge scheme[10] (see Section 9.4.4). The prequantization step performed by his method is based on the activity measure defined by Equation 9.20. Colors are then categorized as belonging

to low, medium, or high activity classes according to two thresholds (t_1 and t_2) determined experimentally. The luminance activity measure (Equation 9.21) is used in complement to color activity to decompose the set of colors with low color activity measures into two subsets corresponding to low and high luminance activity. This decomposition is based on an additional threshold, g. Based on this color classification, the image color set is decomposed into a set of cubes with length $l = 2$, 4, or 8, all colors of one cube belonging to a same class (see Figure 9.12). Therefore, the merge process tends to avoid the merger of clusters associated with low activity measures, while colors associated with high activities are already grouped into larger clusters.

Balasubramanian uses also color activity measures to weight the distance between clusters. Using the same scheme as the aggregation of the activity measure on colors (see Equation 9.20), Balasubramanian defines the color activity of one cluster as the minimal activity of its colors. Therefore, given two clusters C_i and C_j with color activities $\hat{\alpha}_{C_i}$ and $\hat{\alpha}_{C_j}$, the color activity of the merged cluster $\hat{\alpha}_{C_i \cup C_j}$ is equal to $\min(\hat{\alpha}_{C_i}, \hat{\alpha}_{C_j})$. The weight $w_{i,j}$ of two clusters is then defined as

$$w_{i,j} = \frac{1}{\hat{\alpha}^2_{C_i \cup C_j}}$$

To attach large weights to the distance between clusters with small activities, and vice versa, Balasubramanian defines the weighted distance between clusters C_i and C_j as $w_{i,j}\Delta_{i,j}$ where $\Delta_{i,j}$ is the increase of the total squared error induced by the merge of clusters C_i and C_j (Equation 9.14). Using such weights, clusters with high activity measures are more likely to be merged before low activity clusters.

Balasubramanian and Bouman combined their methods[8] so as to integrate the color weights defined by Balasubramanian within the tree-structured vector quantizer defined by Bouman. This method defines the aggregate weight of each cluster as the average weight of its colors. Because the cluster's weights are more meaningful for small and close clusters, Balasubramanian et al. perform $(2/3)K$ unweighted binary splits (Section 9.4.2 and Table 1.1). Then, the $(1/3)K$ remaining splits are performed by splitting,

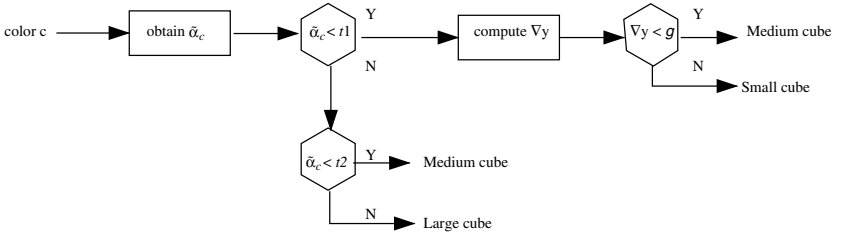


Figure 9.12 Block diagram of prequantization scheme.

at each step, the cluster whose weighted eigenvalue $w_k\lambda_k$ is maximal. The symbols w_k and λ_k denote, respectively, the weight and the maximal eigenvalue of cluster C_k.

9.6 Post-clustering methods

Quantization strategies presented so far use a *pre-clustering* scheme that computes the set of representatives only once. Another strategy consists of defining an initial set of representatives and improving it iteratively. Such a quantization scheme is called a *post-clustering* strategy. The pre-clustering strategy is commonly used, whereas the post-clustering one is less popular. Indeed, the use of a post-clustering strategy generally induces three main drawbacks.

1. First, because the number of iterations is unbounded, quantization algorithms using a post-clustering scheme may be computationally intensive.
2. A much bigger disadvantage generally is the loss of the structure induced by the original strategy, which often makes the mapping step very computationally intensive.
3. Last, the iterative improvement of the set of representatives generally converges to the local minimum the closest to the initial solution.

Therefore, such algorithms are often trapped in local minima of the optimized criterion. However, several heuristics presented in this section have been proposed to improve the computational efficiency of post-clustering algorithms. Moreover, by using a post-clustering scheme, information such as the interrelationships between neighboring clusters and the spatial distribution of colors may be naturally integrated into quantization algorithms. Finally, one common heuristic consists of combining the pre- and post-quantization schemes as follows: First, an initial set of representatives closed from the optimal solution is determined by a prequantization algorithm. This initial set of representative is then moved to the global optimum by a quantization algorithm based on a post-clustering scheme.

9.6.1 The LBG and k-means algorithms

Let us first introduce the LBG algorithm. The design of locally optimal vector quantizers was investigated by both Llyod[47] in 1982[46] and by Linde et al. in 1980. The initialization step defines one set of representatives. This could be the K most commonly occurring colors or any set of colors chosen arbitrary. The iteration begins by assigning each input color to one representative according to some distortion measure. These assignments define a partition of the image color set into clusters, each cluster being associated with one representative. The set of representatives is then modified to minimize the errors relative to input colors. This two-step process is iterated until no

significant change occurs within the set of representatives between two successive iterations.

Several quantization processes, such as those proposed by Heckbert[35] and Braudaway,[15] use the LBG algorithm to improve the set of representatives obtained from their quantizers. However, because the LBG algorithm converges only to a local minimum, this step often yields only slight improvements.[58]

Several investigations have been performed to improve the performance of this algorithm, such as the one by Goldberg,[32] who extends this algorithm to a high-resolution solution that selects a set of representative vectors from input vectors. Likewise, Feng[24] has extended this algorithm to a vector quantizer that exploits the statistical redundancy between the neighboring blocks to reduce the bit rate required by the algorithm.

More significant improvements can be obtained from other heuristics that attempt to select the palette through a sequential splitting process while reducing the TSE at each step. Several investigations such as those of Orchard and Bouman[13] or the one of Balasubramanian[8] propose various splitting procedures and different selection criteria to reduce the TSE.

The k-means algorithms belong to the same category of post-clustering techniques. These algorithms are based on an iterative process that is repeated until it converges to a local minimum solution, i.e., until the cluster centers of the K generated clusters do not change from one iteration to the other. The number of iterations required by the algorithm depends on the distribution of the color points, the number of requested clusters, the size of the color space, and the choice of initial cluster centers.[70] Consequently, for a large clustering problem, the computation can be very time consuming. Recently, faster clustering approaches have been proposed. Among them are some that are commonly used in color image quantization,[56] although these approaches were originally developed for color image segmentation applications, such as the C-means,[19] the fuzzy C-means,[43] and the genetic C-means[56] clustering algorithms or the hierarchical merging approach.[7]

9.6.2 The NeuQuant neural-net image quantization algorithm

The NeuQuant neural-net image quantization algorithm has been developed to improve the common median cut algorithm (Section 9.4.2). This algorithm operates using a one-dimensional, self-organizing Kohonen neural network, typically with 256 neurons, which self-organizes through learning to match the distribution of colors in an input image. Taking the position of each neuron in RGB space gives a high-quality colormap in which adjacent colors are similar.

By adjusting a sampling factor, the network can produce either extremely high-quality images slowly or good images within reasonable times. With a sampling factor of 1, the entire image is used in the learning phase; with a factor of, e.g., 10, a pseudo-random subset of 1/10 of the pixels is used in the learning phase. A sampling factor of 10 gives substantial speed enhance-

ment with a small quality penalty. Careful coding and a novel indexing scheme are used to make the algorithm efficient. This confounds the frequent observation that Kohonen neural networks are necessarily slow.

9.6.3 *The local K-means algorithm*

The local K-means algorithm (LKM), introduced by Verevka and Buchanan,[68,69] is an iterative post-clustering technique that approximates an optimal palette using multiple subsets of image points. This method is based on a combination of a K-means quantization process and a self-organizing map (or Kohonen neural network). The aim of this method is to simultaneously minimize both the TSE and the standard deviation of squared error of pixels,

$$\sigma = \frac{\sqrt{\sum_{\mathbf{c} \in C} (\|\mathbf{c} - q(\mathbf{c})\| - E(C))^2}}{card(C)}$$

Meanwhile, small values of the TSE guarantee that a quantization process accurately represents colors of the original image, and the minimization of the standard deviation preserves variations of colors in the quantized image. The main limitation of this method comes from the two employed measures that treat each pixel independently. Consequently, the spatial correlation among colors is not taken into account. The main advantage of this method is its ability to select a palette without making any assumptions about the boundaries of color clusters.

9.7 *Mapping methods*

The inverse colormap operation is the process that maps an image into a limited set of representative colors. These representatives may be defined by a quantization algorithm or imposed by the default colormap of the output device. To minimize the visual distortion between the input image and the output one, inverse colormap algorithms map each color c of the input image to its nearest representative $\boldsymbol{Q}(c)$. The function $\boldsymbol{Q}$ may be defined by

$$\begin{aligned} C &\rightarrow \{c_1, \ldots, c_K\} \\ c &\rightarrow \boldsymbol{Q}(c) = \arg\min_{z \in \{c_1, \ldots, c_K\}} \|z - c\| \end{aligned} \tag{9.22}$$

where C represents the set of colors of the input image, and $\{c_1, \ldots, c_K\}$ the set of representative colors. The expression $\|z - c\|$ denotes the Euclidean norm of the three-dimensional vector z–c computed in a given color space.

The value $\boldsymbol{Q}(c)$ for a given color c may be computed using an exhaustive search of the minimum of $\|z - c\|$ for all representative colors. Given a 256 $\times$ 256 image and a colormap of 256 colors, this trivial algorithm requires more

than 16 million distance computations (see Equation 9.22). Thus, despite its simplicity and the fact that this method provides an exact solution, it is not practical for large images or interactive applications. Several methods described below have thus been designed to optimize the search of the closest representative. The complexity of the main algorithms is outlined in Table 9.2.

Table 9.2 Complexity of the Main Inverse Colormap Methods

Algorithm	Exact Computation?	Complexity	
		Inverse Colormap Step	Preprocessing Step
Trivial method	Yes	$O(K\|I\|)$	
$k-d$ tree	Yes	$O(\|I\|\log(K))$	$O(K\log(K))$
Locally sorted search	Yes	$O(L\|I\|)$	$O(NK + NL\log(L))$
Three-dimensional Voronoï	No	$O(\|I\|)$	$O(2^{15}\log(K))$
Two-dimensional Voronoï	No	$O(7\|I\|)$	$O(2\|I\| + \|V_I\|)$

Note: Symbols $|I|$ and K denote, respectively, the size of the input image and the number of representatives. Symbols L and N are defined in Section 9.7.4, and symbol $|V_I|$ is defined in Section 9.7.6.

9.7.1 Improvements of the trivial inverse colormap method

Improvements of the trivial inverse colormap algorithm are numerous. Poskanzer[52] proposed improving the search by using a hash table, which allows one to avoid the search of the nearest representative for any color already encountered. However, this optimization remains inefficient for images with a large set of different colors, such as outdoor scenes. One other approach consists of approximating the L_2 Euclidean norm by a less expensive distance metric. Chaudhuri et al.[20] proposed the L_α norm as an approximation of the Euclidean distance, with the L_α norm of a color $\mathbf{c}$ being defined by

$$\begin{aligned}\|\mathbf{c}\|_\alpha &= (1-\alpha)\|\mathbf{c}\|_1 + \alpha\|\mathbf{c}\|_\infty \\ &= (1-\alpha)\sum_{j=1}^{3}|\mathbf{c}^j| + \alpha \max_{i\in\{1,2,3\}}|\mathbf{c}^j|\end{aligned}$$

According to experiments performed by Verevka,[68] the $L_{1/2}$ norm significantly speeds up the search without introducing a noticeable loss in output image quality.

The search can be further reduced by using the following considerations:[36]

- *Partial sum.* Using the square of the L_2 distance, the L_1 norm, or the $L_{1/2}$ norm, the distance between the input color and one representative is defined as a sum of three terms. The partial sum should be compared to the current minimal distance before each new addition. The distance calculation terminates if the partial sum is greater than the current minimal distance.
- *Sorting on one coordinate.* Let us assume that the representatives are sorted along one coordinate axis (e.g., the first one). Then, the search starts with the representative whose first coordinate is the closest to the input color and continues in the increasing first coordinate distance order. The process terminates when the first coordinate distance between the next representative and the input color is greater than the current minimal distance. Note that one should use an axis with a large variance to accelerate the termination of the search.
- *Nearest neighbor distance.* Given a representative $\mathbf{c}_j$ with $j \in \{1, \ldots, K\}$, its closest representative $\mathbf{c}_{q(j)}$ is defined by

$$q(j) = \arg \min_{k \in \{1, \ldots, K\}, k \neq j} d(\mathbf{c}_j, \mathbf{c}_k)$$

Let us suppose that the current representative $\mathbf{c}_i$ traversed by the inverse colormap algorithm is the current closest representative from the input color $\mathbf{c}$. We additionally suppose that the current minimal distance $\Sigma_{min} = d(\mathbf{c}, \mathbf{c}_i)$ is less than one half of $d(\mathbf{c}_i, \mathbf{c}_{q(i)})$. We thus have

$$\begin{cases} \Sigma_{min} = d(\mathbf{c}, \mathbf{c}_i) \\ \Sigma_{min} \leq \frac{1}{2} d(\mathbf{c}_i, \mathbf{c}_{q(i)}) \end{cases}$$

Then, given any other representative color $\mathbf{c}_k$,

$$\begin{aligned} 2\Sigma_{min} \leq d(\mathbf{c}_i, \mathbf{c}_{q(i)}) &\leq d(\mathbf{c}_i, \mathbf{c}_k) \leq d(\mathbf{c}_i, \mathbf{c}) + d(\mathbf{c}, \mathbf{c}_k) \\ \Rightarrow 2\Sigma_{min} &\leq \Sigma_{min} + d(\mathbf{c}, \mathbf{c}_k) \\ \Rightarrow \Sigma_{min} &\leq d(\mathbf{c}, \mathbf{c}_k) \end{aligned}$$

The inverse colormap algorithm should then terminate, as no representative may be closer to $\mathbf{c}$ than $\mathbf{c}_i$.

9.7.2 *Inverse colormap algorithms devoted to a specific quantization method*

As mentioned in Section 9.1 (see also Figure 9.1) any quantization algorithm requires an inverse colormap step. The set of representatives defined by quantization algorithms is built so as to minimize a quantization error defined within some color space (e.g., CIELUV or *YIQ*). On the other hand, inverse colormap algorithms map each color to its nearest representative using an Euclidean norm also defined within a given color space. Therefore, the inverse colormap algorithm should use the same color space as the quantization algorithm to be consistent with the criterion used to design the set of representatives.

The assignment of each color to its nearest representative by inverse colormap algorithms induces a partition of the image color set by a three-dimensional Voronoï diagram defined by the representatives. The partition defined by pre-clustering quantization methods generally does not constitute a Voronoï diagram but may be used as a close approximation of it if the partition has a low total squared error. Moreover, using the partition defined by quantization algorithms, data structures generated by these algorithms may be used to reduce the computation cost of the mapping function. For example, using a tree-structured vector quantizer (Section 9.4.2), the binary tree generated by the quantization algorithm may be used to retrieve the representative of an input color with approximately $O(\log_2(K))$ operations.

9.7.3 *Inverse colormap operations using k – d trees*

Inverse colormap methods may also be studded within the more general nearest neighbor framework. Friedman[11] proposed an algorithm based on an optimized $k-d$ tree to determine the m nearest neighbors of a given color. Given a set of colors, the $k-d$ tree is built by a recursive bipartitioning scheme (Section 9.4.2), each cluster being split along its coordinate axis with the greatest variance so as to have an equal number of colors in both sub-clusters. The recursive split algorithm terminates when each leaf of the binary tree generated by the algorithm contains less than a given number b of colors.

The search procedure initializes a list of the m closest colors encountered so far. Whenever a color is examined and found to be closer than the most distant member of this list, the list is updated. If the node under investigation is not terminal, the recursive procedure is called for the sub-cluster enclosing the query color. When control returns, a test is made to determine if it is necessary to consider the other sub-cluster. This test checks if the box delimiting the sub-cluster overlaps the ball centered at the query color with radius equal to the distance to the mth closest color so far encountered. This test is referred to as the *bounds overlap-ball* test. If the bounds overlap-ball test fails, none of the colors on the other sub-cluster can be among the m closest neighbor of the query color. If the bounds do overlap the ball, then the colors

of the other sub-cluster should be examined, and the procedure is called recursively on this sub-cluster.

The above algorithm may be applied to the inverse colormap framework, by setting $m = 1$ and using the set of representative as the initial set of colors encoded by the $k - d$ tree. According to Friedman, the closest representative is determined in $O(\log_2(K))$ operations.

9.7.4 *The locally sorted search algorithm*

Heckbert's locally sorted search algorithm[35] uses the same basic idea as the bounds overlap-ball test but, instead of performing a recursive decomposition of the color set, Heckbert defines a uniform decomposition of the cube into a lattice of N cubical cells. Each cell of this lattice contains the list of representatives that could be the nearest representative of a color inside the cell. Each cell's list is defined by computing the distance r between the representative closest from the center of the cell and the farthest corner of the cell (see Figure 9.13). This distance gives an upper bound on the distance of any color in the cell to its nearest representative. Therefore, any representative that has a distance to the cell greater than r may be rejected from the list.

Given an input color **c**, Heckbert's method determines which cell contains **c** and traverses its associated list to determine its closest representative. The complexity of this method depends on the mean size of the representative list, which is determined by the number and distribution of representative colors and by the number N of cells composing the lattice. If we denote by L the mean size of the representative lists, the complexity of the preprocessing step that defines the lattice of N cells and sorts the representative lists is equal to $\boldsymbol{O}(NK + NL(L))$.[35] Experiments performed by Heckbert show that the number of distance computations required to determine the representative $\mathbf{Q}(c)$ of a color c may be decreased by 23 for 256 representative

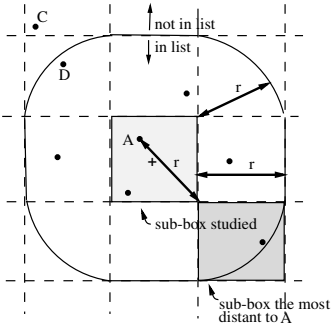


Figure 9.13 The locally sorted search algorithm. The representative A is the closest from the center of the studded sub-box. The distance from A to the farthest corner of the sub-box defines the distance r. The representative D belongs to the list associated with this sub-box, while C is r units away from the sub-box and is thus rejected from the list.

colors and 512 cells. Nevertheless, with a fixed number N of cells, this method remains approximately linear with respect to the number K of representatives. Moreover, the optimal number N of cells is difficult to estimate.

The locally sorted search shows the greatest advantage over an exhaustive search when K is large and when the colors in the input image have a wide distribution.[35] In these cases, the preprocessing time to create the database is overshadowed by the savings in search time.

Some variants of this algorithm, such as those of Houle and Dubois[37] and Goldberg,[32] have been proposed.

9.7.5 *Inverse colormap operation using a three-dimensional Voronoï diagram*

Thomas's method[60] computes the values of the function $\boldsymbol{Q}$ based on a three-dimensional discrete Voronoï diagram defined by the representative colors of the colormap. A discrete Voronoï diagram defined by p points is a partition of a discrete image into a set of p cells, all pixels of one cell being closer to one representative than the others. One advantage of discrete Voronoï diagrams as compared with real ones is that they can be computed, using incremental methods that initialize the n-dimensional image to be partitioned. Thus, using a discrete Voronoï diagram, the cell enclosing a given pixel is retrieved by reading its index in the n-dimensional image.

Using the Thomas method, the Voronoï diagram is encoded with the help of a three-dimnensional array of integers. Each entry of this array represents a color and contains the index of its nearest representative. The main advantage of Thomas's method is that, once the three-dimensional array encoding the three-dimensional Voronoï diagram has been computed, the representative of any input color may be retrieved without any computation. This method is thus quite adapted to mapping any color to the default colormap of a screen. However, it has several drawbacks. First, this algorithm computes the representative of each displayable color. Thus, it involves many useless computations when applied to a few images. Second, using the RGB color space, computation of the three-dimensional Voronoï diagram requires the storage of 256^3 indexes. To reduce the number of initialized values and the amount of required memory, Thomas removes the three least significant bits of each R, G, and B component. The three-dimensional Voronoï diagram is then computed on a $32 \times 32 \times 32$ cube. This heuristic significantly decreases the amount of required memory but introduces several quantization errors (Section 9.2). Moreover, many quantization methods[14,70,74] use color spaces such as CIELUV, CIELAB,[59] or YCrCb,[10] which are more adapted to human vision than the RGB one. In this case, the inverse colormap algorithm has to use the same color space (Section 9.7.2). Using Thomas's method and a different color space, we have to allocate and initialize a three-dimensional image that encloses the transformation of the space. This constraint reinforces the problems linked to the number of data that must be allocated and initialized.

9.7.6 *Inverse colormap operation by a two-dimensional Voronoï diagram*

The basic idea of this method proposed by Brun[18] approximates the three-dimensional image color set by one of its two-dimensional projections so as to perform the inverse colormap operation with a two-dimensional Voronoï diagram instead of a three-dimensional one. Experiments performed by Ohta[48] show that up to 99% of the information (see Equation 9.3) of an image color set is contained in the plane defined by the first two eigenvectors of the covariance matrix.

Given an image color set, Brun computes the first two eigenvectors of the covariance matrix (Equation 9.2) and defines a projection operator p onto the plane P_{princ} defined by these two eigenvectors. The three-dimensional Voronoï diagram associated with the colormap $\{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$ is then approximated by a two-dimensional diagram V_I defined by the sites $\{p(\mathbf{c}_1), \ldots, p(\mathbf{c}_K)\}$.

The mapping of a color $\mathbf{c}$ to the representative whose associated cell in V_I encloses $p(c)$ induces two successive approximations. First, the three-dimensional distances are approximated by two-dimensional ones. This first approximation neglects the variations of the dataset along the third eigenvector. Second, the use of a discrete Voronoï diagram involves rounding of each projected color $p(\mathbf{c})$ to the nearest pixel in V_I.

Due to these successive approximations, this first mapping function often fails to map the input colors to their closest representatives. However, these approximations cause minor errors in the distance computations, so the errors often affect the indices of adjacent Voronoï cells. The correction of the approximations thus requires us to compute the neighborhood of each Voronoï cell. This neighborhood may be encoded by using the dual of the Voronoï diagram V_I, named a Delaunay graph[6] and denoted by D_I. The data structures V_I and D_I are then combined in the following manner.

Given an input color $\mathbf{c}$, Brun reads in V_I the index $V_I[p(c)]$ of the cell enclosing $p(c)$. The set $D_I[V_I[p(c)]]$ containing $V_I[p(c)]$ and the index of the cells adjacent to it is then read from the Delaunay graph D_I. The color $\mathbf{c}$ is then mapped to its closest representative among $D_I[V_I[p(c)]]$.

$$Q(c) = \arg \min_{i \in D_I[V_I(p(c))]} \|\mathbf{c} - \mathbf{c}_i\|^2 \tag{9.23}$$

where $\mathbf{c}_i$ denotes a representative color of the colormap $\{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$.

Note that Equation 9.23 uses three-dimensional distances, the two-dimensional Voronoï diagram being used only to restrict the number of distance computations. Therefore, this method is similar to the one of Heckbert (Section 9.7.4), which performs a partition of the color space and associates a list of representatives to each cell of the partition. The complexity of this inverse colormap method is determined by the mean size of lists $D_I[V_I[p(c)]]$ (see Equation 9.23), which is related to the mean number of neighbors of a two-

dimensional Voronoï cell. It can be shown[53] that this number is bounded by six. Therefore, once the diagrams V_I and D_I are computed, this method requires less than seven distance computations per pixel. The overall complexity of the method is bounded by $O(9|I|+|V_I|)$ where $|I|$ and $|V_I|$ denote, respectively, the size of the input image and the array V_I.

9.8 Dithering methods

Until recently, digital halftoning was commonly considered to be a process of displaying continuous tone images on bilevel output devices (e.g., printers).[34,39,41,50] To extend these methods to color images, several generalizations have been proposed.

Considering that the capability of the human visual system to distinguish different colors declines rapidly for high spatial frequencies, digital halftoning has been exploited to enhance *a posteriori* quantization processes by adding complementary perceived colors. Indeed, the human visual system may create additional perceived colors by averaging colors in the neighborhood of each pixel. Thus, halftoning trades off high spatial resolution in favor of increased color resolution. Several investigations[33] have shown that excellent image quality with color palettes of very small size (as few as four colors) can be obtained when digital halftoning is used. A variety of halftoning techniques have been proposed to reach this objective, such as error diffusion techniques, dithering techniques, and model-based halftoning techniques.

The common problem of these approaches is that they consider each color component to be an individual grayscale image; consequently, due to the correlation between the color components, color shifts and false textural contours appear on the resulting image.[4,58,67] This problem is all the more noticeable when the number of colors that forms a palette is very small. That is why some investigations have been proposed to develop vector error diffusion techniques to reduce effects induced by the correlation between color components.

To optimize the performance of dithering techniques, some adaptive techniques[4,66] have been also investigated. The key idea of these techniques is to compensate quantization errors by using an image-adaptive cost function. More recently, other approaches have combined dithering and quantization processes or designed quantizers to match the requirements of dithering algorithms.

9.8.1 Error diffusion methods

The object of error diffusion is to quantize the image in such a way that the average value of the quantized image in a local region is the same as the average value of the original one in the same local region. In error diffusion, any errors produced in the quantization process in a pixel are propagated to neighboring pixels to be negated subsequently (see Figure 9.14) so that, in a small group of neighboring pixels, the average color is more accurate.

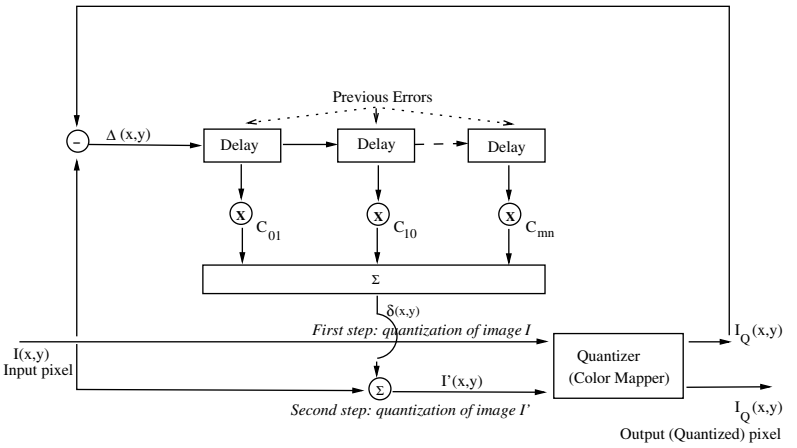


Figure 9.14 The dithering algorithm. Input pixels are dithered by the weighted sum of the previous quantization errors before being quantized for the second time.

Let $I'(x, y)$ be the pixel value resulting of the pseudo-random process, and $\Delta(x, y) = I(x, y) - I_Q(x, y)$ be the quantization error at pixel location (x, y); then, $I'(x, y)$ can be described by

$$I'(x, y) = I(x, y) + \delta(x, y)$$

with

$$\delta(x, y) = \sum_{i=0}^{m} \sum_{j=0}^{n} C_{ij} \Delta(x - i, y - j)$$

which represents a two-dimensional spatial filter (see Figure 9.15), such that

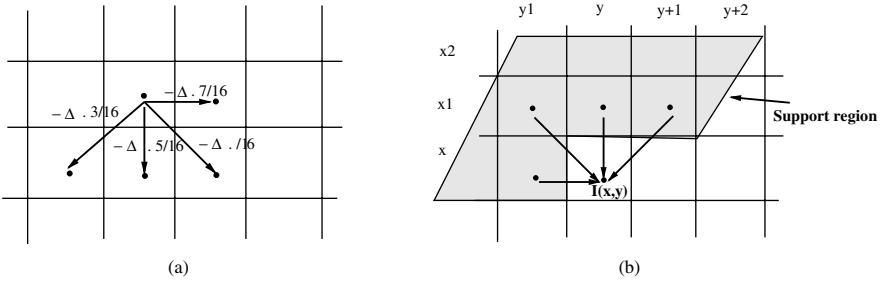


Figure 9.15 The Floyd–Steinberg algorithm: (a) the propagation of the quantization error to neighboring pixels and (b) the contribution of neighboring pixels to the noise vector.

$$C_{00} = 0 \text{ and } \sum_{i=0}^{m} \sum_{j=0}^{n} C_{ij} = 1$$

where the indexes i and j define the neighborhood over which quantization errors are accumulated according to coefficients C_{ij}.

The constraints on C_{ij} ensure that, locally, the quantization error averages to a zero value. Because the objective of error diffusion is to preserve the average value of the image over local regions, a unity-gain, lowpass, finite impulse response (FIR) filter is used to distribute the error. Because the eye is less sensitive to high spatial frequencies, the image resulting from an error diffusion process typically appears closer to the original image than those obtained with ordered dither[58] (Section 9.8.2).

This algorithm was originally proposed by Floyd and Steinberg[27] in 1975. Several variants have been proposed, such as those proposed by Fletcher,[27] Heckbert,[35] Bouman,[12] Dixit,[22] and Akarun.[3] Several improvements have also been proposed to speed the process, such as the parallel implementation proposed by Fletcher[25] and the fuzzy technique proposed by Akarun.[1] According to Fletcher,[22] this optional pixel correction function is usually necessary when the number of output colors is very small, e.g., 32 or fewer.

Several investigations have shown that the error diffusion algorithm presents some major drawbacks. For example, according to Fletcher[27] and Bouman,[12] the colormap selection algorithm may be altered by the dithering process when input colors lie outside the convex hull of the representative colors. It is thus necessary to guarantee that all colors in the original image may be generated by taking a linear combination of colors in the colormap. Likewise, according to Knuth,[41] another major drawback of the error diffusion algorithm is its inherent serial property; e.g., the value $I'(x, y)$ depends in all pixels of input data $I(x, y)$. Furthermore, it sometimes puts "ghosts" into the picture. Even if the ghosting problem can be ameliorated by choosing the coefficients C_{ij} so that their sum is less than 1, the ghosts cannot be exorcised completely in this way. Finally, according to Bouman,[12] when the error diffusion algorithm is used in conjunction with an unstructured color palette, it requires an exhaustive search of the closest representative (Section 9.7). Although some basic improvements to the exhaustive search method have been made (Section 9.7.1), this method remains computationally intensive.

9.8.2 Ordered dither methods

The ordered dithering methods add non-random patterns $\delta(i, j)$ to blocks of pixels $I(x, y)$ before the quantization step. The quantized image $I'(x, y)$ is thus deduced from the original one by

$$I'(x, y) = I(x, y) + \delta(x \bmod n, y \bmod n)$$

where $\delta(i, j)$ is an $n \times n$ matrix of dithering values. The matrix δ is designed to have a 0 average value and an energy spectrum with a minimal energy at low spatial frequencies.[41] The amplitude of the pattern is determined so that any area of constant color value is quantized into a variety of nearby color values, thereby breaking up regions of false contouring. This reduces the correlated error patterns that are characteristic of areas of false contouring.

In contrast to random dithering processes, the basic idea of ordered dither techniques is to use a finite, deterministic, and localized threshold function to decrease the correlation of quantization errors. Likewise, in contrast to random dithering processes, ordered dither techniques are image-adaptive; i.e., these techniques exploit the local characteristics of the image to obtain improvements over a constant filter.

Several investigations have shown that the ordered dither algorithm presents some major drawbacks. According to Bouman,[12] the application of ordered dithering is complicated by the nonuniform spread of colors within the palette. As the distance between nearby colors may vary significantly across the color space, it may be impossible to determine the amplitude that will sufficiently dither all areas of constant color value in the image without adding noticeable noise to other areas. Bouman et al.[12] have thus proposed some adjustments to improve this technique. Nevertheless, it has been shown that no ordered dither method guarantees that, after quantization, the error spectrum will remain concentrated at high spatial frequencies.

According to Kim et al.,[40] ordered dither methods have the disadvantage of producing a binary-recursive and computerized texture that is unfortunately unsuitable. Meanwhile, the Floyd–Steinberg method has the disadvantage of producing intrusive and snake-like patterns that are also unsuitable. They thus proposed the dot diffusion method, which avoids both of these unsuitable properties and combines the sharpness of the Floyd–Steinberg method with the parallelism of the order dither method. According to Kim et al., dot diffusion methods have the desired property but tend to blur the image in the edge regions through the diffusion of the quantization error. This last problem is solved by an activity-weighted dot diffusion method that divides the dithering process into two sub-processes, one dependent on the object frequencies and one independent.

Another approach to an ordered dithering technique has been proposed by Lin et al.[45] to enhance performance of compression schemes such as joint bilevel group (JPIG) techniques. The interest of this approach, based on a pixel interleaving (i.e., grouping pixels with similar dithering thresholds) strategy is that it could be also extended to quantization scheme.

9.8.3 Vector dither methods

The major drawback of scalar diffusion techniques is that the color components are analyzed independently while they are highly correlated. Indeed, the natural ordering used by scalar quantizers is lost within the vector quantizer framework. Imposing an ordering, or organization, on a vector

quantizer requires careful considerations. Rather than defining an ordering on vectors, some investigations proposed a two-stage approach. In the first stage, the original image is quantized by a scalar process, and an error propagation mask is applied to compensate the quantization errors. In the second stage, the color palette is ordered, and the resulting image is once again quantized by a vector process based on the ordered palette. Finally, an error propagation mask is used one more time to compensate for the quantization errors. This approach has been used by Goldschneider et al.[33] to propose an embedded multilevel error diffusion technique.

Another strategy of vector error diffusion has been proposed by Akarun et al.[4] In their paper, the authors proposed two new adaptive diffusion techniques, respectively based on vector and scalar diffusion, to correct the disturbing effects of error diffusion strategies. The proposed adaptation criterion is based on the statistics of the quantization error obtained from the dithered image. Rather than considering each color component as an individual grayscale image to obtain a scalar diffusion, Akarun et al.[4] proposed using either the Karhunen–Loeve color coordinate system or a pseudo-Karhunen–Loeve system, which provides better results than the two other scalar diffusion techniques.

9.8.4 Joint quantization and dithering methods

Instead of performing *a posteriori* the dithering process to eliminate contouring effects or other disturbing effects, some investigations have combined the dithering and quantization processes. Because the error diffusion step changes pixel values after the quantization step, the set of representative designed by the quantizer is obviously not optimal in regard to the corrected image.

A first attempt to design the set of representatives based on a dithering process has been proposed by Akarun et al.[2] Akarun's method combines a tree-structured vector quantizer (Section 9.4.2) and a dithering process based on a controlled displacement of cluster centers. The key idea is to obtain a wider colorspace so as to obtain more quantization colors in the color palette and hence eliminate color impulses and false contours.

Another strategy consists of processing a joint optimization of the selection of the representatives and their locations to benefit from the spatial averaging performed by the human visual system (HVS). Indeed as mentioned in Sections 9.5 and 9.8, one notable characteristic of the HVS is the rapid decline of its ability to distinguish different colors for high spatial frequencies. This phenomenon creates additional perceived colors by a visual local averaging of the representatives. This effect is widely used by dithering techniques but cannot be readily incorporated in usual quantization algorithms, because the set of representatives is known only after the quantization step.

A joint optimization of the selection of the representatives and their locations has been proposed by Scheunders et al.[57] This method combines a

quantization process based on a competitive learning clustering technique and a dithering process based on a simple local error diffusion technique. To take the error diffusion process into account, an objective error function is first computed and next minimized based on a competitive learning scheme that alternatively updates the palette colors in the color space and the color pixels in the color image.

Other authors[26,54] use a model of the spatial averaging performed by the HVS within the joint quantization scheme. This spatial averaging may be modeled by three lowpass filters,[26,51] $(W_k)_{k \in \{1, \ldots, N\}}$. Let us denote by $S_{i,k}$ the spatial support of filter W_k at position i. Given an image $I = (I_i)_{i \in \{1, \ldots, N\}}$, the kth component of the perceived image $\tilde{I}$ is given by

$$\forall k \in \{1, 2, 3\} \forall i \in \{1, \ldots, N\} \qquad \tilde{I}_k(i) = \sum_{j \in S_{i,k}} W_k(j) I(j)_k$$

Given a set of representatives $\{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$, the location of each representative may be encoded by an $N \times K$ Boolean matrix such that $M(i, j)$ equals 1 if the pixel I_i is mapped to $\mathbf{c}_j$, and 0 otherwise. Using the $K \times 3$ vector R of representatives such that $R(i, k)$ is the kth coordinate of $\mathbf{c}_i$, the output image I_Q is defined as

$$\forall i \in \{1, \ldots, N\} \qquad I_Q(i) = M(i, .)R$$

where $M(i, .)$ denotes the ith line of matrix M.

The perceived output image $\tilde{I}_Q$ is then given by

$$\forall k \in \{1, 2, 3\} \forall i \in \{1, \ldots, N\} \qquad \tilde{I}_Q(i)_k = \sum_{j \in S_{i,k}} W_k(j) M(j, .) R(., k)$$

where $R(., k)$ denotes the kth column of matrix R. Note that $I_Q(i)_k = M(i, .)R(., k)$.

The distance between the perceived input and output images may then be defined as the squared Euclidean distance between perceived colors,

$$\begin{aligned} H(M, R) &= \sum_{k=1}^{3} \sum_{i=1}^{N} (\tilde{I}(i)_k - \tilde{I_Q}(i)_k)^2 \\ &= \sum_{k=1}^{3} \sum_{i=1}^{N} (\Sigma_{j \in S_{i,k}} W_k(j) I(j)_k - \Sigma_{j \in S_{i,k}} W_k(j) M(j, .) R(., k))^2 \end{aligned}$$

The minimization of the error $H(M, R)$ thus requires us to set both the matrices M and R. A minimization of $H(M, R)$ using a fixed value of R has been proposed by several halftoning methods.[26,51] However, a joint

minimization of M and R has been proposed only recently by Puzicha,[54] who uses an alternating minimization scheme: (1) minimize $H(M, R)$ with respect to M, keeping R fixed; and (2) minimize $H(M, R)$ with respect to R, leaving M fixed. This two-step strategy is iterated until convergence. Each optimization step is performed by an annealing method whose convergence is accelerated by a hierarchical decomposition of the image.

9.9 Conclusion and perspectives

This chapter has surveyed the current research on color quantization. Prospectively, significant gains in image quality may be obtained through further research on processing algorithms and display techniques, and considerable improvements should result from research on color metrics designed for complex image scenes instead of the CIE metrics, which are based on large uniform areas. That is, only a few papers take into account the perceptual aspects of color within quantization algorithms. Two strategies have nevertheless been investigated to extend the use of perceptual aspects within quantization algorithms.

The first strategy consists of using either a perceptually uniform or a luminance/chrominance color space. The main characteristic of a uniform color space is that the perceived difference between two colors is proportional to the Euclidean distance between their representations in the uniform color space. A number of such spaces (e.g., CIELAB color space) that achieve this objective to varying degrees have been used in color image quantization.[29] Most uniform color spaces are also based on a luminance/chrominance separation. Some investigations[29,38,54] studied the influence of the color space on the quality of the quantized images. However, none of these studies exhibited one color space as having definitive advantages within the quantization framework.

The second strategy consists of using spatio-color parameters that enable us either (a) to compensate *a posteriori* the main visible degradations between the original image and the quantized one or (b) to minimize *a priori* the number of visible degradations that may appear on the quantized image with regard to the original image. The strategy that minimizes *a priori* the spatio-color distortions should obviously be the best one. However, quantization algorithms using this strategy are generally computationally intensive. Moreover, the design of analytic measures taking into account all visual distortions of a quantized image is still an open problem.

With image context playing an important role in human color vision, the use of a uniform color space is not sufficient to quantify perceptual color distance. Likewise, the relative positions of different colors in the image plane greatly influence our color interpretation. To handle this characteristic of the human visual system, several approaches, such as those described in Section 9.5, have integrated a color activity criterion into the quantization scheme. Unfortunately, such approaches remain highly heuristic and are far from offering a mathematical model for context dependency of colors that

can be integrated as an objective function to a quantization process.[40,74] In this context, color appearance models appear promising. However, most of them are fairly complex, and their suitability for color imaging remains to be comprehensively evaluated.[58] The development of simple models, more adapted to complex image scenes, is a promising challenge for color imaging applications.[62]

References

1. Akarun, L., Ozdemir, D., and Alpaydin, E., Fuzzy error diffusion of color images, *IEEE Processing*, 46–49, 1997.
2. Akarun, L., Ozdemir, D., and Yalcin, O., Joint quantization and dithering of color images, *Proc. IEEE, ICIP'96*, 557–560, 1996.
3. Akarun, L., Yardimci, Y., and Cetin, A. E., Adaptive methods for dithering color images, *Proc. IEEE*, 125–128, 1995.
4. Akarun, L., Yardimci, Y., and Cetin, A. E., Adaptive methods for dithering color images, *IEEE Trans. Image Processing*, 6(7), 950–955, 1997.
5. Anderberg, M. E., *Cluster Analysis for Applications*, Academic Press, New York, 1973.
6. Aurenhammer, F., Voronoï diagrams: a survey of fundamental geometric data structure, *ACM Computing Surveys*, 33(3), 345–405, 1991.
7. Balasubramaian, R. and Allebach, J., A new approach to palette selection for color images, *J. Imaging Technol.*, 17(6), 284–290, 1991.
8. Balasubramaian, R., Allebach, J., and Bouman, C. A., Color-image quantization with use of a fast binary splitting technique, *J. Opt. Soc. Am.*, 11(11), 2777–2786, 1994.
9. Balasubramaian, R., Bouman, C. A., and Allebach, J., Sequential scalar quantization of color images, *J. Electronic Imaging*, 3(1), 45–59, 1994.
10. Balasubramaian, R. and Allebach, J., A new approach to palette selection for color images, *Human Vision, Visual Processing, and Digital Display III (1991)*, SPIE 1453, 58–69, 1991.
11. Bentley, J. L., Friedman, J. H., and Finkel, R. A., An algorithm for finding best matches in logarithmic expected time, *ACM Trans. Math. Software*, 3, 209–226, 1977.
12. Bouman, C. and Orchard, M., Color image display with a limited palette size, *Visual Communications and Image Processing IV (1989)*, SPIE 1199, 522–533, 1989.
13. Bouman, C. and Orchard, M., Color quantization of images, *IEEE Trans. Signal Processing*, 39(12), 2677–2690, 1991.
14. Braquelaire, J. P. and Brun, L., Comparison and optimization of methods of color image quantization, *IEEE Trans. Image Processing*, 6(7), 1048–1052, 1992.
15. Braudaway, G., A procedure for optimum choice of a small number of colors from a large color palette for color imaging, *Proc. Electronic Imaging '86,* Boston, MA, November 1986, 75–79.
16. Brun, L., *Segmentation d'images couleur à base Topologique*, Ph.D. thesis, Université Bordeaux I, 351 cours de la Libération 33405 Talence, December 1996.
17. Brun, L. and Mokhtari, M., Two high speed color quantization algorithms, *Proc. CGIP'2000*, 116–121, Saint Etienne, October 2000.
18. Brun, L. and Secroun, C., A fast algorithm for inverse color map computation, *Computer Graphics Forum*, 17(4), 263–271, 1998.

19. Celenk, M., A color clustering technique for image segmentation, *Computer Vision, Graphics, and Image Processing*, 52, 1990, 145–170.
20. Chaudhuri, C. A., Chen, W. T., and Wang, J., A modified metric to compute distance, *Pattern Recognition*, 7(25), 667–677, 1992.
21. Chou, P. A., Lookabaugh, T., and Gray, R. M., Optimal pruning with applications to tree-structured source coding and modeling, *IEEE Trans. Inf. Theory*, 2, 299–315, 1989.
22. Dixit, S. S., Quantization of color images for display/printed on limited color output devices, *Comput. Graphics*, 15(4), 561–568, 1991.
23. Equitz, W. H., A new vector quantization clustering algorithm, *IEEE Trans. Acoustics, Speech, and Signal Processing*, 37(10), 1568–1575, 1989.
24. Feng, Y. S. and Nasrabadi, N. M., Dynamic address-vector quantization of RGB colour images, *IEEE Proc.*, 138(4), 225–231, 1991.
25. Fletcher, P., A SIMD parallel colour quantization algorithm, *Comput. Graphics*, 15(3), 365–373, 1991.
26. Flohr, T. J., Kolpatzik, B. W., Balasubramanian, R., Carrara, D. A., Bouman, C. A., and Allebach, J. P., Model based color image quantization, *Human Vision, Visual Processing, and Digital Display IV (1993)*, SPIE 1913, 270–281, 1993.
27. Floyd, R. W. and Steinberg, L., An adaptive algorithm for spatial gray scale, in SID, Ed., *Int. Symp. Dig. Tech. Papers*, 36, 1975.
28. Froidevaux, C., Gaudel, M. -C., and Soria, M., *Types de Données et Algorithmes*, McGraw-Hill, New York, 1990.
29. Gentile, R., Allebach, J., and Walowit, E., Quantization of color images based on uniform color spaces, *J. Imaging Technol.*, 16(1), 11–21, 1990.
30. Gersho, A., and Gray, R. M., *Vector Quantization and Signal Compression*, Kluwer Academic, Norwell, MA, 1991.
31. Gervautz, M. and Purgathofer, W., A simple method for color quantization: octree quantization, in N. Magnenat-Thalmann and D. Thalmann, Eds., *New Trends in Computer Graphics*, Springer-Verlag, New York, 1988, 219–231.
32. Goldberg, N., Colour image quantization for high resolution graphics display, *Image and Vision Computing*, 9(1), 303–312, 1991.
33. Goldschneider, J. R., Riskin, E. A., and Wong, P. W., Embedded mutilevel error diffusion, *IEEE Trans. Image Processing*, 6(7), 956–964, 1997.
34. Gomes, J. and Velho, L., Image processing for computer graphics, in *Digital Halftoning*, Springer Verlag, New York, 1997.
35. Heckbert, P. S., Color image quantization for frame buffer display, *ACM Computer Graphics (ACM SIGGRAPH '82 Proc.)*, 16(3), 297–307, 1982.
36. Hodgson, M. H., Reducing the computation requirements of the minimum distance classifier, *Remote Sensing of Environment*, 25, 117–128, 1988.
37. Houle, G., and Dubois, E., Quantization of color images for display on graphics terminals, *Proc. IEEE Global Telecom. Conf. (GLOBE_COM '86)*, December 1986, 284–297.
38. Jain, A. K. and Pratt, W. K., Color image quantization, *National Telecom. Conf. 1972 Record*, December 1972.
39. Jarvis, J. F., Judice, C. N., and Ninke, W. H., A survey of techniques for the display of continuous tone pictures on bilevel displays, *Comput. Graphics Image Processing*, 4, 13–40, 1976.
40. Kim, K. M., Lee, C. S., Lee, E. J., and Ha, Y. H., Color image quantization and dithering method based on visual system characteristics, *J. Imaging Sci. Technol.*, 40(6), 502–509, 1996.

41. Knuth, D. E., Digital halftones by dots diffusion, *ACM Trans. Graphics*, 6(4), 245–273, 1987.
42. Kurz, B. J., Optimal color quantization for color displays, *IEEE Computer Vision and Pattern Recognition Proc.*, January 1983, 217–224.
43. Lim, Y. W. and Lee, S. U., On the color image segmentation algorithm based on the thresholding and the fuzzy C-means techniques, *Pattern Recognition*, 23(9), 935–952, 1990.
44. Lin, J., Storer, J., and Cohn, M., On the complexity of the optimal tree pruning for source coding, in *Proc. of Data Compression Conf.*, IEEE Computer Society Press, Los Angeles, CA, 1991, 63–72.
45. Lin, Y., Wang, Y., and Fan, T. H., Compaction of ordered dithered images with arithmetic coding, *IEEE Trans. Image Processing*, 10(5), May 2001, 797–802.
46. Linde, Y., Buzo, A., and Gray, R., An algorithm for vector quantizer design, *IEEE Trans. Commun.*, COM-28, 1, 84–95, 1980.
47. Llyod, S. P., Least squares quantization, *IEEE Trans.*, IT-28, 129–137, 1982.
48. Ohta, Y.-I., Kanade, T., and Sakai, T., Color information for region segmentation, *Computer Vision, Graphics, and Image Processing*, 13, 222–241, 1980.
49. Paeth, A. W., Mapping RGB triples onto four bits, in A. S. Glassner, Ed., *Graphics Gems,* Academic Press, Cambridge, MA, 1990, 233–245, 718.
50. Pappas, T., Model-based halftoning of color images, *IEEE Trans. Image Processing*, 6(7), 1014–1024, 1997.
51. Pappas, T., Model based halftoning of color images, *IEEE Trans. Image Processing*, (7) 1014–1024, 1997.
52. Poskanzer, J., PBM+ image processing software package, 1991.
53. Preparata, F. P. and Shamos, M., *Computational Geometry,* Springer-Verlag, New York, 1985.
54. Puzicha, J., Held, M., Ketterer, J., Buhmann, J. M., and Fellner, D. W., On spatial quantization of color images, *IEEE Trans. Image Processing*, 9(4), 666–682, 2000.
55. Rogowitz, B., The human visual system: a guide for display technologist, *Proc. of the SID*, 24, 235–252, 1983.
56. Scheunders, P., A genetic approach towards optimal color image quantization, *IEEE Proc.*, 1031–1034, 1996.
57. Scheunders, P. and Backer, S. D., Joint quantization and error diffusion of color images using competitive learning, in *IEEE Proc.*, ICIP-97, 1, 811–814, 1997.
58. Sharma, G. and Trussell, H. J., Digital color imaging, *IEEE Trans. Image Processing*, 6(7), 901–932, 1997.
59. Tajima, J., Uniform color scale applications to computer graphics, *Computer Vision, Graphics, and Image Processing*, 21(3), 305–325, 1983.
60. Thomas, S. W., Efficient inverse color map computation, in J. Arvo, Ed., *Graphics Gems II*, Academic Press, Cambridge, MA, 1991, 116–125, 528–535.
61. Trémeau, A., *Analyse d'images couleurs: du pixel à la scène*, Ph.D. thesis, Université Jean Monnet, Habilitation à Diriger des Recherches, 1998.
62. Trémeau, A., Color contrast parameters for analysing image differences, *Color Image Science 2000 Conf. Proc.,* Univ. of Derby, April 2000, 11–23.
63. Trémeau, A., Calonnier, M., and Laget, B., Color quantization errors in terms of perceived image quality, in *Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, ICASSP '94, 5, Adelaide, Australia, April 1994, 93–96.

64. Trémeau, A., Charrier, C., and Cherifi, H., A vector quantization algorithm based on the nearest neighbor of the furthest color, *Proc. IEEE Int. Conf. on Image Processing,* ICIP '97, Santa Barbara, CA, October 1997, 3, 682–685.
65. Trémeau, A., Dinet, E., and Favier, E., Measurement and display of color image differences based on visual attention, *J. Imaging Sci. Technol.*, 40(6), 522–534, 1996.
66. Trémeau, A. and Laget, B., Color dithering based on error diffusion and subjective measures of image quality, in *Proc. IASTED Int. Conf. Signal and Image Processing*, Las Palmas, CA, February 1998, 359–362.
67. Verevka, O., Digital halftoning, http://web.cs.ualberta.ca/oleg/dithering.html, 1995.
68. Verevka, O. and Buchanan, J., Local k-means algorithm for color image quantization, *Proc. Graphics Interface '95,* Quebec, Canada, May 1995, 128–135.
69. Verevka, O. and Buchanan, J., Local k-means algorithm for color image quantization, *Proc. Graphics Interface '95,* Quebec, Canada, May 1995, 128–135.
70. Wan, S., Wong, S., and Prusinkiewicz, P., An algorithm for multidimensional data clustering, *ACM Trans. Mathematical Software*, 14(2), 153–162, 1988.
71. Wan, S. J., Prusinkiewicz, P., and Wong, S. K. M., Variance-based color image quantization for frame buffer display, *Color Res. Appl.*, 15(1), 52–58, 1990.
72. Wong, S., Wan, S., and Prusinkiewicz, P., Monochrome image quantization, in *Proc. Canadian Conf. on Electrical and Computer Engineering*, September 1989, 17–20.
73. Wu, X., Optimal quantization by matrix searching, *J. Algorithms*, 12(4), 663–673, 1991.
74. Wu, X., Color quantization by dynamic programming and principal analysis, *ACM Trans. Graphics*, 11(4), 348–372, 1992.
75. Wu, X. and Zhang, K., A better tree-structured vector quantizer, in *Proc. IEEE Data Compression Conf.*, IEEE Computer Society Press, Los Angeles, CA, 1991, 392–401.
76. Xiang, Z., Color image quantization by minimizing the maximum intercluster distance, *ACM Trans. Graphics*, 16(3), 1997.
77. Xiang, Z. and Joy, G., Color image quantization by agglomerative clustering, *IEEE Computer Graphics and Applications*, 14(3), 44–48, 1994.

chapter ten

Gamut mapping

Ján Morovic
University of Derby, U.K.

Contents

10.1 Introduction

The ability to reproduce color images on various imaging media (e.g., computer and television displays, print, or projection screens) is required in an ever increasing range of contexts. Examples include displaying web content on a given computer's display, viewing a DVD on a television screen, printing some holiday pictures, and projecting a business presentation. Furthermore, the images to be reproduced can come from a variety of sources such as the surrounding environment, image data captured with digital cameras, conventional photographs, or original artwork in either analog or digital form.

As the media mentioned above can, and often do, have different achievable ranges (gamuts) of colors, it is frequently the case that some colors cannot be made to match the original exactly. For example, the appearance of some bright green colors that can be displayed on a television in a dark room cannot be achieved in a printed newspaper, so we can say that the gamuts of these two media do not match. As a result of such *gamut mismatches*, it becomes necessary to alter the original colors to ones that a given medium is capable of reproducing. How this replacement, which is frequently referred to as *gamut mapping*, is to be done so as to end up with a good reproduction of the original image is the question that this chapter intends to discuss and for which it intends to provide a framework. While the above kind of gamut mismatch is also an issue for reproducing single colors, there is a simple solution to that (i.e., the original color is reproduced by the closest reproducible color), so we will here focus exclusively on the gamut mapping of complex images.

Unlike many previous discussions about gamut mapping, which are surveys of work carried out related to this topic and which present its various aspects in a general way,[41] this chapter intends to approach the subject from the point of view of what happens to an image when it is reproduced between a pair with media with different gamuts. The reason for doing this is to provide the reader with a clear framework for implementing a gamut-mapping solution. Furthermore, this will be done by frequent reference to an example cross-media reproduction scenario that is introduced early in

this chapter, and relevant parts of the theory will be discussed and illustrated. A further feature of this chapter is that it relies heavily on material covered in other parts of this book, to which the reader will be directed for details of transformations that an image needs to undergo both before and after gamut mapping itself.

Before introducing the example scenario, however, it is necessary to define some key terms. While the reader will be familiar with many of these, it is still important to define them as used in this chapter (for more rigorous definitions, see Reference 46).

The first key term is that of an *image*, which here will be understood to mean a two-dimensional visual stimulus. Associated with this is the concept of *digital image data,* which are two-dimensional arrays of value multiplets containing information about an image. In other words, while an image is something that is by definition visible, digital image data are not.

Next, it is necessary to introduce the term *color imaging medium,* which here will stand for a medium for capturing or displaying color information (e.g., scanners, digital cameras, displays, prints, projections). *Color imaging devices* are then devices that bring about color reproduction media and can either be identical to them (e.g., a display is both a device and a medium) or different from them (e.g., a printer is a device that is used for obtaining a print — the medium).

Finally, a *color gamut* is the range of a set of colors. Because these can be represented as locations in a three-dimensional color space, a color gamut can be represented as a volume in such a space. Given that a gamut is a volume and that this volume is finite, it also has a surface, and this is referred to as the *gamut boundary.* Examples of sets of colors whose gamuts are of interest are the colors in an image or the colors that are reproducible within a given medium.

The example scenario we will work with in this chapter involves an image on a cathode-ray tube (CRT) display, which has a white point simulating *CIE Standard Illuminant D50* (CIE, 1981). We want to obtain a printed reproduction of it using a color laser printer (Figure 10.1) whereby this

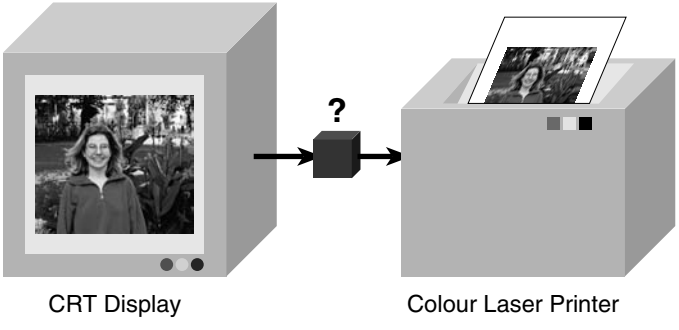


Figure 10.1 Example cross-media color reproduction scenario.

reproduction will be seen in a viewing cabinet using a D50 simulator light source. Furthermore, the chromaticities of the display are set to be very close to the chromaticities of the printed substrate as seen in the viewing cabinet, and both white points have a luminance of approximately 100 cd/m^2. This is done so as to exclude chromatic adaptation and dynamic response differences between original and reproduction conditions and thereby focus more clearly on the gamut mismatch. Note that, while all details of this scenario are from an actual image displayed on a particular display and reproduced using a particular printer, the commercial details of the devices involved will not be revealed, as they are not relevant to this chapter. Many more aspects of this scenario will be introduced or determined as we progress through the chapter.

Before looking at the issues related to overcoming the gamut mismatch between our original and reproduction media, we will consider a number of preliminary topics in the following section.

10.2 Preliminary issues

10.2.1 Reproduction intent

First, we need to consider the desired properties of the reproduction. This is the basis for making decisions for all of the various aspects of the reproduction system that will exist between the original and reproduction media. (In Figure 10.1, this is represented by the black box.) The specification of what properties we want in the reproduction (or what properties it actually ends up having) is customarily referred to as a *color rendering intent* or *color reproduction objective*, and there are a number of ways to define it.

Hunt[20] introduced the following six color reproduction objectives:

- *Spectral reproduction.* The spectral power distributions of original and reproduction should match.
- *Exact reproduction.* The relative luminances, chromaticities, and absolute luminances should match.
- *Colorimetric reproduction.* The chromaticities and relative luminances should match.
- *Equivalent reproduction.* The chromaticities and relative and absolute luminances of the original should appear to match the reproduction.
- *Corresponding reproduction.* The chromaticities and relative luminances in the reproduction should appear to match the original when both have the same luminance levels.
- *Preferred reproduction.* Appearance match should be sacrificed to achieve a more pleasing result.

Another way of defining reproduction objectives is to specify only two main types: *accurate reproduction* and *pleasant reproduction*.[40] Accurate reproduction

is characterized by the reproduced image appearing as close to the original image as possible with respect to gamut differences, and pleasant reproduction aims to ensure that the reproduced image is pleasant in isolation. Note that this implies that accurate reproduction aims to preserve the pleasantness level of the original (i.e., unpleasant originals are reproduced as unpleasant and pleasant originals are reproduced as pleasant), whereas pleasant reproduction aims to make the reproduction pleasant regardless of the original's pleasantness.

Returning to Hunt's reproduction objectives, it can be seen that the first five objectives set targets for the relationship between colors in the original and the reproduction in colorimetric terms, and they define various degrees of accurate reproduction. The sixth (preferred) reproduction objective is then identical to pleasant reproduction in that it sets an objective for the reproduction that does not refer to the original and instead specifies that the reproduction needs to have a certain property in isolation — i.e., to look pleasant.

Another important set of rendering intents was defined for the purposes of color management systems by the ICC and consists of the four enumerated below.[21]

1. *Perceptual intent.* "The exact gamut mapping of the perceptual intent is vendor specific and involves compromises such as trading off preservation of contrast in order to preserve detail throughout the tonal range. It is useful for general reproduction of images, particularly pictorial or photographic-type images."
2. *Saturation intent.* "The exact gamut mapping of the saturation intent is vendor specific and involves compromises such as trading off preservation of hue in order to preserve the vividness of pure colors. It is useful for images which contain objects such as charts or diagrams."

Next, there are two colorimetric intents that "preserve the relationships between in-gamut colors at the expense of out-of-gamut colors. Mapping of out-of-gamut colors is not specified but should be consistent with the intended use of the transform."

3. *Media-relative colorimetric intent.* "This intent rescales the in-gamut, chromatically adapted tristimulus values such that the white point of the actual medium is mapped to the white point of the reference medium (for either input or output). It is useful for colors that have already been mapped to a medium with a smaller gamut than the reference medium (and therefore need no further compression)."
4. *ICC-absolute colorimetric intent.* "For this intent, the chromatically adapted tristimulus values of the in-gamut colors are unchanged. It is useful for spot colors and when simulating one medium on another (proofing). Note that this definition of ICC-absolute colorimetry is

actually called 'relative colorimetry' in CIE terminology, since the data has been normalized relative to the illuminant."

Again, the last two of these rendering intents specify variants of accurate reproduction, whereas the first two are forms of pleasant reproduction. Furthermore, it is possible to think of cases where a hybrid combination of accurate and pleasant reproduction is required. For example, an image showing both objects having corporate identity colors and people might be reproduced so that accuracy is the aim for the former and pleasantness for the latter.

In the context of our scenario, we too need to decide the reproduction objective; we will try to achieve *accurate reproduction*. The reason is that accurate reproduction has been studied extensively, and there is relatively little recent research that looks at pleasant color reproduction, which is a significantly more complex task than accurate reproduction. In a sense, pleasant color reproduction should be considered only after accuracy can be achieved.

10.2.2 Original and reproduction data and intermediate color spaces

Having decided the rendering intent, we can now proceed to specifying a color reproduction system that can be used for pursuing it. This system will have the original image's data as its input, and it should produce data for the reproduction medium that, when rendered on it, will result in an image that satisfies the rendering intent. In our scenario, the reproduction should be as close to the original's appearance as possible.

Before looking at how to transform image data, it is necessary to look at what will be transformed. We need to consider the data we have for the original image and the data we need for the reproduction medium. In addition, as both of these descriptions of the image are medium specific (or device dependent), we also need to decide what intermediate color space to use. An intermediate color space is used for the reasons discussed in Chapter 4 (i.e., to reduce the number of transformations that must be computed for communicating images between a number of media) and because of the requirements of gamut mapping in terms of the space in which it is to be preformed.

10.2.2.1 Original data

In our scenario, the original image is displayed on a CRT display and as such will be available in a digital file containing red, green, and blue (RGB) values for each picture element (pixel). Sending these digital image data to the display results in an image. Because we intend to reproduce the appearance of this image, it is necessary to quantify it. This can be done by first using a characterization model to compute tristimulus values for each pixel. Then, we must take into account the influence of viewing conditions so as to predict the appearance attributes (e.g., lightness, chroma, and hue) corresponding to the tristimulus values of each pixel. For this, a color appearance model is used.

10.2.2.2 *Intermediate color space*

The obvious choice of intermediate color space is one based on a color appearance model that takes viewing conditions into account. However, it is necessary to ensure that the intermediate color space used in a color reproduction system is suitable for gamut mapping, and this might require further transformations of image data before gamut mapping is carried out. Most gamut mapping algorithms (GMAs) intend to preserve some appearance attributes of an original color while changing others (e.g., preserve hue and adjust lightness and chroma so as to move an original color into the reproduction gamut). It is therefore important to have a color space whose dimensions accurately represent the appearance attributes with which a given GMA intends to work. In this context, we need to discuss two important issues.

1. What appearance attributes should a GMA use?
2. What color space predicts those attributes most accurately?

In terms of the first issue, there are two options: first, *lightness, chroma,* and *hue,* and, second, *brightness, colorfulness,* and *hue* (see Chapter 3). Which of these to use will depend on the specifics of the rendering intent, which in turn shows that rendering intents need to be specified more narrowly than was done in Section 10.2.1. From this point of view, it is important to specify to what extent the chromaticity and absolute intensity of the light source under which the original is seen should be taken into account. If appearance is to be reproduced in an absolute way, then brightness and colorfulness are used; if the original's appearance relative to the adopted white is of importance, then lightness and chroma are used. In most cases, the latter is chosen, which will also be the case for our scenario. An example of where the former could be appropriate is art reproduction, where one might want to maintain the appearance of a painting as seen in its original surroundings, including the effect of the original viewing conditions' intensity and possible color cast.

The second issue relating to color space choice needs to be considered very carefully. Because most GMAs intend to work with appearance attributes, and because this can be done only in terms of a color space, it is important that the values in a color space accurately predict appearance attributes. If a given algorithm intends to maintain hue, then this is implemented by maintaining the hue predictor of a given color space; if this hue predictor is not accurate; then the reproduction will have unwanted properties. Imagine a highly chromatic blue color in the original and an algorithm that intends only to reduce its chroma and leave lightness and hue unchanged. To do this, it is necessary to have a color space whose appearance predictors are accurate, otherwise, changing the chroma predictor's value might also result in changes of other color appearance attributes (even though those attributes predictors' values would be left unmodified). If such a transformation is done in the CIELAB color space, then the resulting color's

hue would change from blue to purple, even though CIELAB hue angle would be maintained.

Note therefore that this limitation of color spaces (that they predict appearance attributes imperfectly), and the fact that GMAs work in terms of their dimensions, means that the algorithms described in subsequent parts of this chapter can deal only with the predictors and not with the actual appearance attributes. Hence, an algorithm that intends to preserve hue in effect preserves only the hue predictor of the color space in which it is implemented (e.g., CIELAB's h^*_{ab} or CIECAM97s' h).

As a result of this situation, a number of recent studies have evaluated the degree to which various color spaces (in particular, CIELAB[7] and CIECAM97s[29]) predict given color appearance attributes. The deficiencies of hue predictors have been described by Hung and Berns[19] and Ebner and Fairchild,[11] and these studies show that none of the current color spaces predicts hue well for all parts of color space. A number of papers propose solutions to these imperfections of existing color spaces. Ebner and Fairchild,[12] for example, describe the IPT color space developed so as to improve hue uniformity, Marcu[36] describes the mLAB space developed for the same reason, McCann[37] discusses additional general shortcomings of CIELAB, and Zeng[60] proposes a gamut mapping solution that uses different color spaces for different parts of color space.

Related to the decision regarding the intermediate color space in which gamut mapping is performed, we need to choose the reference (or adapted) white that is used when calculating its coordinates. It is important to ensure that this choice actually reflects the viewing conditions under which original and reproduction are to be viewed. If they are viewed apart from each other, then their respective medium white points (i.e., the brightest achromatic colors achievable on their media) are the reference whites; alternatively, if they are viewed simultaneously, then the medium white with the higher brightness (or luminance) should be the reference white for both media. A third option is the use of the perfect diffuser as the reference white, which is appropriate if the original and reproduction are to be viewed in an environment that includes other content.

Having decided what intermediate color space to use, what the reference white is, and what appearance attributes to use for the gamut mapping, a GMA can be applied, and this results in such colors being assigned to all pixels that can be achieved in the reproduction medium.

10.2.2.3 Reproduction data

Once color appearance space coordinates are available for the reproduction image's pixels, it is necessary to compute device-dependent data for the reproduction medium. In the case of a print medium, this can either be values for each of the colorants it uses (i.e., in most cases, cyan, magenta, yellow, and black, CMYK) or RGB values (if the printer cannot be controlled directly in terms of its own native data). Again, this transformation consists first of an inverse color appearance or color space transform that provides the

tristimulus values for the pixels, and second of an inverse medium characterization model that computes medium-specific data to produce the specified tristimulus values (see Chapter 5). To summarize the issues discussed in this section, see Figure 10.2, which shows the data discussed here as well as the five-stage transform[31] that it implies.

For our example scenario, we will be using the CIECAM97s color space's lightness, chroma, and hue predictors as the intermediate color space and the medium whites as the adapted whites.

10.3 Color gamuts

Now that we have covered the preliminary issues of what the reproduction should be like and what color space will be used, we can turn to understanding the gamuts that are involved in cross-media color reproduction. As defined previously, a color gamut is the range of a set of colors. Here, we discuss the implications of this definition and introduce methods for describing and visualizing color gamuts. Although the definition of a color gamut is very simple, its implications are often overlooked. Some of them[50] will therefore be made explicit here.

10.3.1 Implications of color gamut definition

The phenomenon of color requires an *observer*, a *stimulus*, and *viewing conditions*. Because this is the case for individual colors, it necessarily is also the case for their ranges — color gamuts. Consequently, it is essential to specify

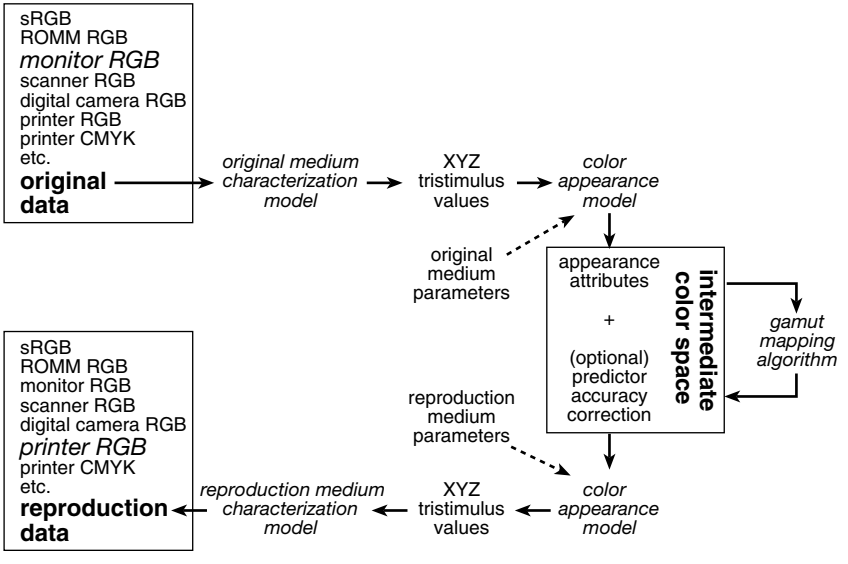


Figure 10.2 Original and reproduction data and the intermediate color space.

the details of all three of these components when talking about a color, and hence also when talking about a color gamut. Therefore, it is also meaningless to talk about the color gamut of a set of stimuli in general (e.g., the gamut of a given CRT). Such a set has a color gamut only when seen by an observer under some viewing conditions.

For example, discussing the color gamut of a printed image is meaningless, as it could assume a number of different forms that are subject to viewing conditions and observers; a printed image has a set of possible color gamuts rather than a single one. Looking at such an image in the dark gives a zero-volume gamut (to use an extreme example); different levels of illumination result in different gamut volumes. Illumination chromaticity changes gamut shape as well as volume, and viewing distance and flare in the environment make a difference, too. Hence, the question, "What is the gamut of this print?," cannot be answered and should instead be rephrased as, "What is the gamut of this print under viewing conditions *X* for observer *Y*?"

10.3.2 Describing color gamut boundaries

Given a set of stimuli viewed by a given observer under a given set of viewing conditions, we can describe their color gamut. The outcome of this will be a *gamut boundary descriptor* (GBD). The calculation of a GBD will consist of analyzing the set of colors it is to describe and of determining their range.

If, for example, we only wanted to describe the gamut boundaries of grayscale images and grayscale imaging media, this problem would be one dimension and have a single solution; i.e., the GBD would consist of the largest and smallest lightnesses either present in an image or achievable by a medium. However, as we are dealing with chromatic as well as achromatic data, the problem of gamut boundary description is a three-dimensional one, and a number of approaches can be used. Before we discuss some methods for gamut boundary description, let us first take a closer look at some sets of colors whose gamuts we might want to compute.

10.3.2.1 Colors of image and media in example scenario

In terms of our example scenario, we need to describe the gamuts of the original image and of the reproduction medium. For some algorithms, we also need to know the gamut of the original medium. To calculate the gamuts of these sets of colors, we can either analyze all of them or deal only with smaller samples. In the original image, there are in excess of 60,000 distinct colors. This number was obtained by calculating the three-dimensional color histogram of the image in CIECAM97s, with a bin size of one unit in each dimension, and counting the number of bins that had pixel frequencies greater than zero. For the two media, on the other hand, the number of colors resulting from all the possible data that can be sent to them is of the order of magnitude of hundreds of thousands. Note that we are not talking about

the number of RGB combinations (of which there are $2^{24} \cong 16.8 \times 10^6$ for the media when controlled in terms of 8 bit/channel RGB) but about the number of distinguishable colors as predicted by some color appearance model.

Given this volume of data, it is useful to take a smaller sample for the purposes of our first view of the gamuts involved here, even though we might want to base our actual gamut boundary calculations on an analysis of all the colors of the image and the media. The sampling that was used for the gamut visualization shown in Figures 10.3 through 10.5 was as follows.

The medium dependent coordinates (i.e., RGB) for the two media were sampled at 10 intervals per channel, resulting in 1000 RGB triplets. For these, CIECAM97s coordinates were calculated by first obtaining XYZ values using medium characterization models and then calculating CIECAM97s lightness (J), chroma (C), and hue (h) predictors for an average surround (a background that had 50% luminance compared to the adapted white's luminance under test conditions). It was further assumed that the original and reproduction media are seen separately, and the adapted white for each medium was therefore that medium's white point. Figures 10.3 and 10.4 show the result of this sampling. An alternative to this kind of sampling strategy (i.e., one that takes samples from the entire volume of the medium's medium-dependent coordinates) is to take samples only from the surface of the gamut in medium-dependent terms. For a given number of samples, this will give a finer sampling of the gamut's surface. A necessary condition for this to result

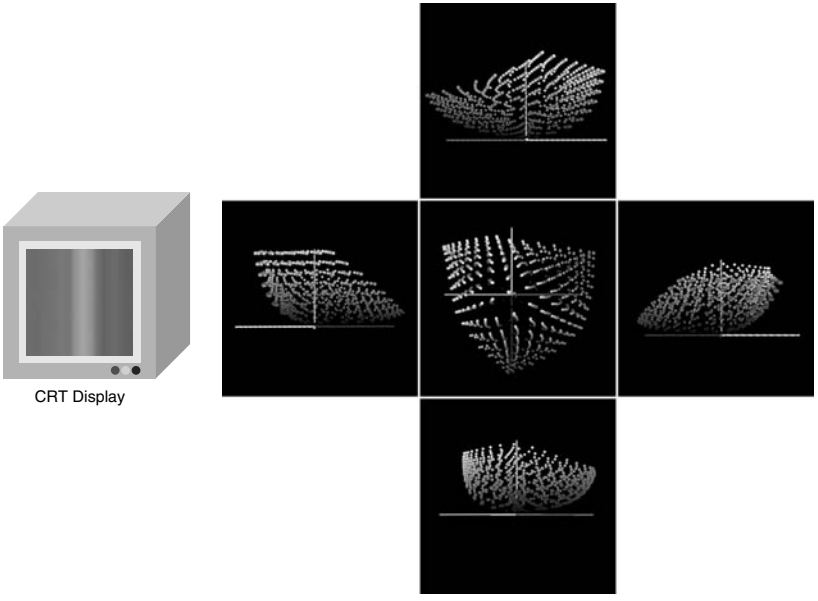


Figure 10.3 ***(See color insert following page 430)*** One thousand samples taken from the CRT's gamut.

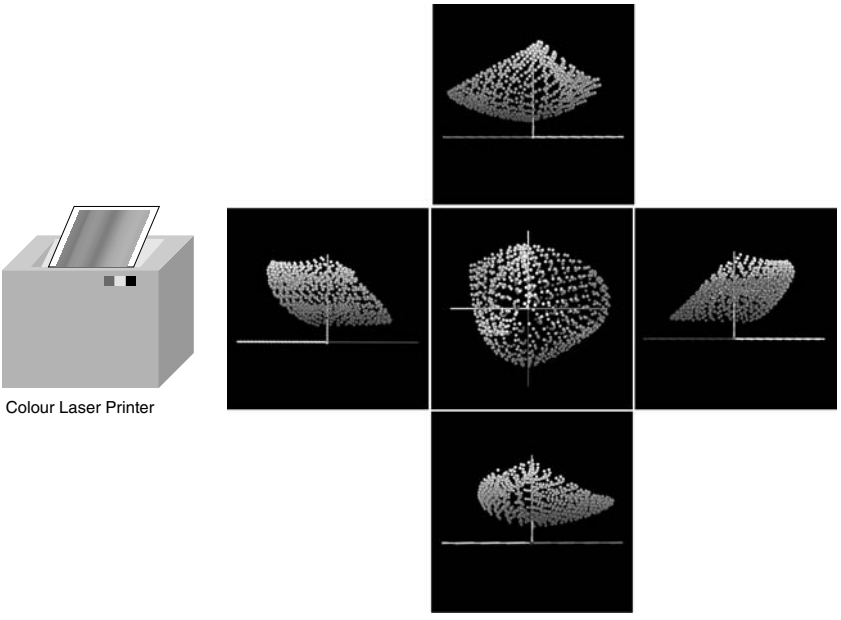


Figure 10.4 ***(See color insert)*** One thousand samples taken from the color laser print's gamut.

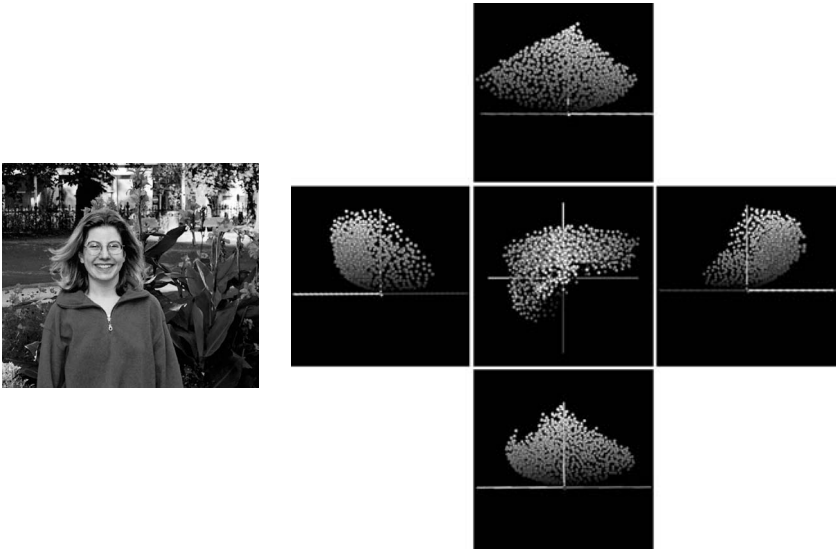


Figure 10.5 ***(See color insert)*** One thousand ninety-one samples taken from the original image's gamut.

in a good representation of the gamut boundary in color appearance space is a monotonic relationship between medium-dependent and color appearance representations.

Note that Figures 10.3 through 10.5 contain views of the colors (shown as spheres that have colors corresponding to their centers' coordinates in color space) sampled from the image and media. Furthermore, each of the five views shown in each figure is the result of what is seen when the various axes of the CIECAM97s JCh color space are orthogonal to what is seen. In fact, the axes shown in red and green are the positive and negative parts, respectively, of the *a* axis. Those shown in yellow and blue are the positive and negative parts of the *b* axis. The *J* axis is shown in gray, and each of these axes (or, more precisely, pairs of half-axes for *a* and *b*) is 100 units in length. Note also that *a* and *b* are not defined in the CIECAM97s model, and they are simply the orthogonal equivalents corresponding to *C* and *h*. They are calculated as $a = C^*\cos(h)$ and $b = C^*\sin(h)$. Whenever there is mention of the *Jab* space in this chapter, what is meant is this orthogonal equivalent of the CIECAM97s *JCh* space.

The sampling of the images' colors was done in terms of their *Jab* coordinates, and the following method was used:

1. Choose the first color in the image as one of the samples.
2. For each subsequent color, check whether it is closer to any of the samples than a given threshold (6 *Jab* units was used here); if it is not, add it to the set of chosen samples.
3. Using this method, 1091 colors were sampled (Figure 10.5) and, as follows from the sampling technique used, no color in the image is farther from a sampled color than 6 *Jab* units.

10.3.2.2 Methods for describing color gamut boundaries

As mentioned previously, the gamuts of sets of three dimensional color coordinates can be described in a number of ways, and these can be generic, medium specific, or image specific.

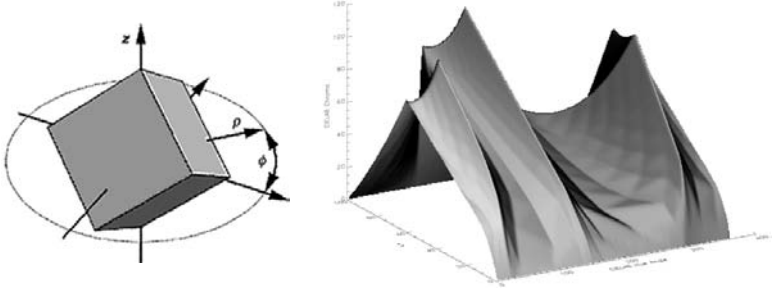
10.3.2.2.1 Medium-specific methods. To begin with, medium-specific methods are either based on extracting a gamut boundary descriptor (GBD) directly from a specific medium characterization model or they exploit some other aspect of the fact that the set of colors whose gamut is to be described is all the colors a given medium can achieve. Examples of the first type of medium-specific gamut boundary description algorithms are the method developed by Engeldrum[13] for calculating GBDs from the Kubelka–Munk equations and Mahy's method[35] for doing this from the Neugebauer equations. The second type of approach includes the solution by Inui[22] based on using partial differential equations to find the surface in color space beyond which changes to the inputs of a characterization model no longer result in three-dimensional changes in a color appearance space. Another example of this kind of approach is Herzog's *gamulyt* method,[15] which exploits the fact

that the gamuts of most media have the shape of a distorted cube — the vertices being black, white, red, green, blue, cyan, magenta, and yellow (Figure 10.6a). However, a feature of all these methods is that they are only suitable for calculating the gamuts of very specific sets of colors, and none of them can be used for calculating image gamut boundaries. That is, these approaches make assumptions about the set of colors whose gamut they can describe that are not applicable to the colors of images; e.g., some of them expect the set to be convex or to have a cube-like shape.

10.3.2.2.2 Generic methods. The development of generic GBD methods was aimed at overcoming the previously described inherent limitation of medium-specific solutions. Such methods include Kress and Stevens's proposal[61] to use convex hulls for the description of gamut boundaries and Cholewo and Love's suggestion[62] to use alpha-spheres for this purpose. Braun and Fairchild[2] then developed the *mountain range* method, which uses gridding and interpolation to arrive at a data structure consisting of a uniform grid in terms of lightness and hue and stores the gamut's most extreme chroma values for each of the grid points (Figure 10.6b). Finally, the *segment maxima* method[45] is based on dividing color space evenly in terms of spherical angles and storing the point with the largest radius for each segment. As this is the method that will be used in our scenario, it will be introduced in detail next.

Using the segment maxima method, the gamut boundary of a color reproduction medium (or an image from it) is described by a matrix containing the most extreme colors for each *segment* of color space. This segmentation can be carried out either in terms of J, C, and h or spherical coordinates whereby spherical coordinates can be calculated from orthogonal Jab coordinates using the following formulæ:

$$r = [(J - J_E)^2 + (a - a_E)^2 + (b - b_E)^2]^{1/2} \tag{10.1}$$



(a) (b)

Figure 10.6 Gamut boundary calculation methods: (a) gamulyt cube and (b) mountain range grid.

$$\alpha = \tan^{-1}((b - b_E)/(a - a_E)) \tag{10.2}$$

$$\theta = \tan^{-1}[(J - J_E)/((a - a_E)^2 + (b - b_E)^2)^{1/2}] \tag{10.3}$$

E is defined as the center of the gamut to be described and can be obtained by averaging the coordinates of the points that will be used for obtaining the GBD (alternatively, the point having *Jab* coordinates of [50, 0, 0] can be used in most cases, as it is a good approximation of the center from the point of view of this method), *r* is the distance of a color from the center, α is an angle in the *ab* plane having a range of 360°, and θ is the angle in a plane of constant α having a range of 180° (Figure 10.7a). Note that the coordinates of *E* can be different for different gamuts and that, even though the method is described here in terms of CIECAM97s, it can easily be adapted for use in other color spaces.

The GBD matrix is calculated by first dividing color space into $n \times n$ segments (e.g., $n = 16$ was used in a previous study[43] and will also be used for the example scenario here) according to either α and θ (Figure 10.7b) or *J* and *h*. Hence, the data are stored in each segment either in terms of α, θ, and *r*, or *h*, *J*, and *C*, respectively. The number of segments to be chosen depends on the accuracy needed, where accuracy increases with *n* given that a sufficient number of samples is available from the gamut to be described. If only smaller numbers of samples are available, increasing *n* will not increase accuracy and can in fact result in artefacts. To calculate a GBD for a set of colors (either sampled from an imaging medium or an image) in terms of spherical angle segmentation, the following procedure can be used:

1. Set up an empty GBD.
2. Transform each of the set of *Jab* values from which we calculate a GBD into spherical coordinates using Equations 10.1 through 10.3.

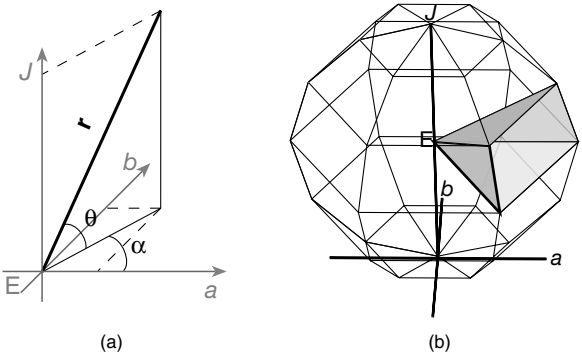


Figure 10.7 Overview of segment maxima GBD in CIECAM97s: (a) spherical coordinates and (b) sphere segmented in terms of α and θ (only 6 × 6 segments — of which one is highlighted — are shown for the sake of clarity).

3. For each of the spherical coordinates from Step 2 do the following:
   a. If the segment is empty, store the current color.
   b. Otherwise, if the radius of the current color is larger than the radius of the color stored for the segment, then store the current color for that segment. Note that it is not only r that is stored for a given segment but spherical angles as well.
   c. Otherwise, ignore the current color.
4. Check whether any segment is left empty. If so, interpolate a value for it based on neighboring occupied segments (this is very rare for color imaging media but can often be the case when image gamuts are calculated).

An important advantage of this method is that the GBD points obtained using it are actual colors from the gamut boundary of the set of samples used, and inaccuracies in the descriptor are due only to the number of segments chosen. Note that the segment maxima method will be used throughout the remainder of this chapter and that Figure 10.8 shows the gamut boundaries calculated using it for the media in the example scenario.

Given a segment maxima GBD, it is also straightforward to calculate the *volume* of a color gamut, and this can be done by summing up the volumes of all the distinct tetrahedra having E and sets of three neighboring points from the GBD as their vertices. For three points from a GBD to be neighbors, they need to be stored in members of the GBD matrix that are next to each other horizontally, vertically, or diagonally (e.g., points with GBD matrix

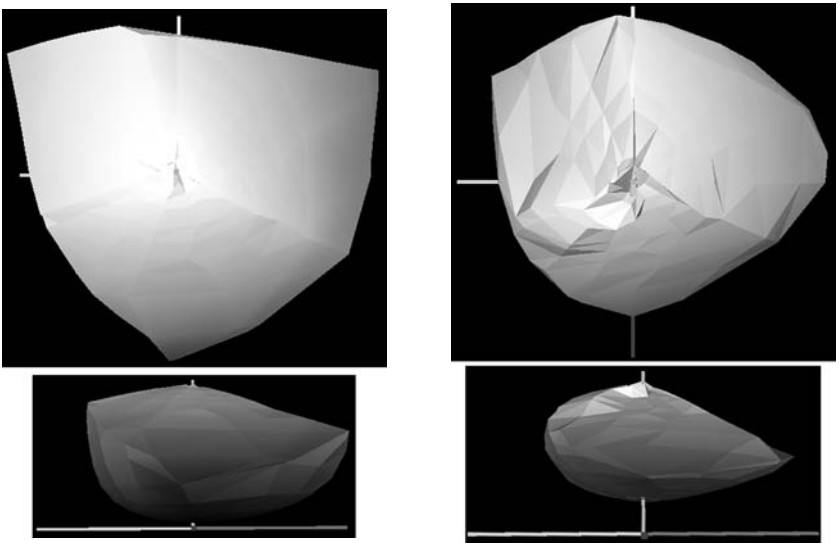


Figure 10.8 ***(See color insert)*** Gamuts of (a) CRT and (b) color laser print from example scenario.

indices [*i*, *j*], [*i* + 1,*j*], [*i*, *j* + 1] are a valid choice, but [*i* – 1, *j*], [*i*, *j*], [*i* + 1, *j*] are not, as [*i* – 1, *j*] and [*i* + 1, *j*] are not neighbors). Also bear in mind that α is circular and that, when data is organized such that the columns relate to α and the rows to θ, the last column of the matrix is a neighbor of the first column.

10.3.2.2.3 Image-specific issues. As far as the description of image gamut boundaries is concerned, a number of additional concerns apply compared to when medium gamuts are considered. While the calculation of the gamut of all the colors in an image is a straightforward task, and one that can be achieved using the generic methods discussed above, it is another matter altogether to calculate the perceived gamut of an image — in other words, a gamut that describes the range of colors that an image is *perceived* to have rather than simply the gamut of all of its pixels.[48] This distinction needs to be made, as images can contain pixels whose individual colors will either not be perceived (e.g., if they are below the human visual system's spatial acuity threshold) or will not fully contribute to the image's perceived gamut (e.g., if they are very infrequent in an image). Imagine, for example, a large mid-gray image that contains one pixel each with the following colors from a medium's gamut boundary: black, white, red, green, and blue. Looking at the image, these pixels might not even be distinguishable from the gray ones, but computing all of the image pixels' gamut would suggest that it is as large as that of the medium on which it is shown.

While there are no established methods for reliably determining the perceived gamuts of images, a number of approaches have been suggested by Morovic and Sun,[48] and they all consist of excluding some of an image's pixels when calculating that image's gamut. The first of these is to exclude colors in an image that occur only rarely, i.e., whose frequency is below a certain threshold. Pixels in high spatial frequency areas should also be excluded, as they play a lesser part in how the image's color range is perceived. However, further research is needed for developing a robust and experimentally justified image gamut description method.

10.3.2.3 Visualizing gamut boundaries

An important part of developing and implementing a cross-media color reproduction system (and therefore a gamut mapping solution) is to visualize the gamuts as well as individual colors used for testing it. While this can be done by producing plots of the data as projected onto or intersected by planes, it is often more useful to have a way of visualizing the data in a manner that allows for easy interaction with it. To do this, a very simple solution is to generate *virtual reality mark-up language (VRML)* models; indeed, that is how most of the gamut plots were obtained for this chapter. While VRML is a complex and extensive language, the visualization of gamuts and colors in color space requires only relatively few shape types to be used. Furthermore, VRML files are nothing but text files, and they can be viewed using freely available plug-ins for web browsers.

To generate VRML files containing some simple shapes all that is needed is to have "#VRML V2.0 utf8" as the first line of a text file, after which, for example, the *box* and *sphere* shapes can be used. For example, to have a box with its center at [XYZ] = [50, 0, 0], [width, depth, height] = [100, 2, 2] and [R, G, B] = [100%, 0%, 0%], the following text needs to appear in the VRML file:

```
Transform {
   translation 50 0 0
   children [
      Shape{
         appearance Appearance {material Material {diffuseColor 1 0 0}}
         geometry Box {size 100 2 2}
      }
   ]
}
```

A sphere with its center at [XYZ]=[0, 100, 0], a radius of 2 and [R, G, B] = [0%, 100%, 0%] can then be had by including the following in the VRML file:

```
Transform {
   translation 0 100 0
   children [
      Shape{
         appearance Appearance {material Material {diffuseColor 0 1 0}}
         geometry Sphere {radius 2}
      }
   ]
}
```

Having more shapes present in a given scene is accomplished simply by adding more of the above kind of code to the VRML file. In fact, Figures 10.3 through 10.5 were generated using only the *box* and *sphere* shapes in VRML whereby the centers of the spheres there had [X, Y, Z] = [*a*, *b*, *J*] coordinates and colors calculated from the *Jab* values using the inverse color appearance model, and then a CRT's characterization model that resulted in the RGB values specified in the *diffuseColor* part of the code. To obtain colored solids and meshes, the *IndexedFaceSet* and *IndexedLineSet* VRML shapes, respectively, can be used. Details of how to do this can be found in any textbook on VRML or the language's specification.

Before we can proceed to the discussion of gamut mapping itself, we still need to cover one more issue related to gamuts themselves: methods for calculating where a given line intersects a gamut boundary.

10.3.3 Line gamut boundary algorithms

Given gamut boundary descriptors of the original and reproduction, a gamut mapping algorithm will try to make changes to the original image

so as to ensure that all its colors end up inside the reproduction gamut. To do this, algorithms often intend to change colors along straight lines in color space on the basis of where the original color and the original and reproduction gamut boundaries are on such a line. Therefore, it is necessary to devise some methods for determining where a given gamut boundary intersects a given line. Such intersections are referred to as *line gamut boundaries (LGBs)*.

In some cases, GBD algorithms have implicit methods for calculating LGBs along lines of certain properties. For example, Braun and Fairchild's *mountain range* method[2] allows for an easy calculation of LGBs for lines of constant hue and lightness, as do Herzog's *gamulyt* method[15] and the *segment maxima* method[45] when using cylindrical segmentation (i.e., segmentation in lightness and hue).

To obtain the intersection of a gamut boundary and other lines, it is possible to use some iterative interpolation techniques with the above methods implicit in GBD algorithms, or the *flexible sequential line gamut boundary (FSLGB)* algorithm introduced by Morovic and Luo[45] can be used to analytically obtain the intersection of any line of constant hue angle and a gamut boundary described using the *segment maxima* method. As this is the method that will be used in our example scenario, it will be described in detail next.

As FSLGB is intended for lines of constant hue angle, it first finds the two-dimensional gamut boundary at the hue angle in which the mapping is to be carried out (Figure 10.9). For a given color (C), this is done in the following way:

1. Calculate equation of constant hue angle plane (φ) having hue angle of C (α_C).
2. For each θ level in the $n \times n$ *segment maxima* GBD, find the pair of neighboring points from the GBD matrix of which one has a larger and one a smaller hue angle than α_C.
3. For each pair, calculate the intersection of the line connecting the two GBD points with φ.
4. In addition to these n points, calculate the points on the J axis where the surface defined by the GBD matrix intersects it.

For the top of the lightness axis, this can be done by considering only the n GBD points from segments having the largest θ values. Triangles are then formed between the point with the largest J and neighboring pairs of the other points. The intersection of each of these triangles and the lightness axis is calculated, and if it is within the triangle, then it is the LGB point. An analogous procedure is used for finding the intersection of the gamut boundary with the bottom of the lightness axis.

The resulting set of $n + 2$ points form a polygon (Figure 10.9c) describing the gamut boundary for a given hue angle. The intersection of a given line (l) and this polygon can then be found using the following procedure:

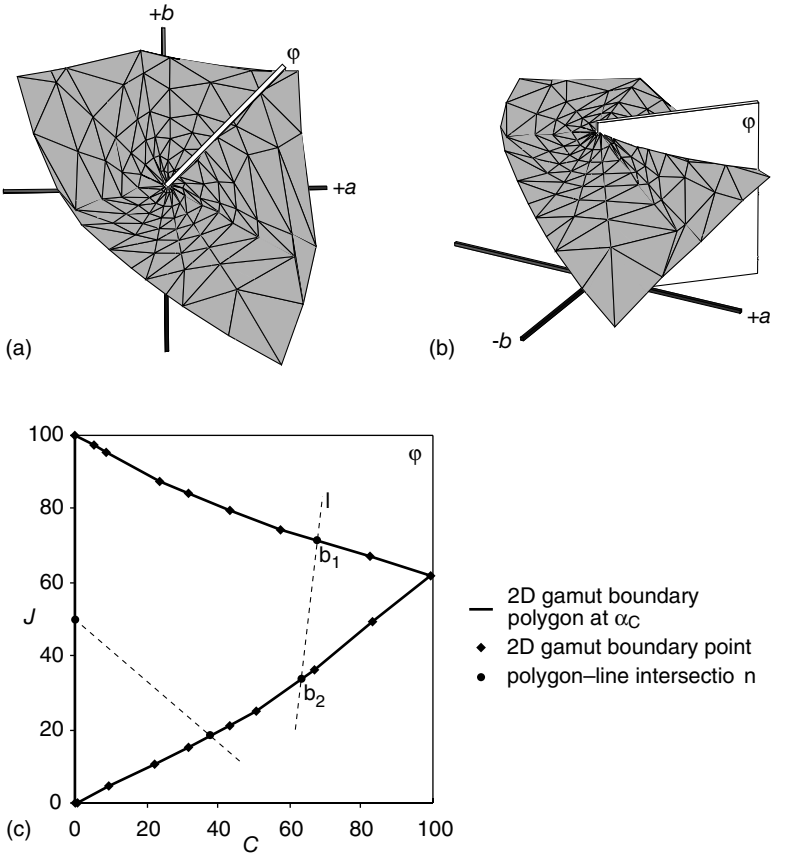


Figure 10.9 Overview of FSLGB algorithm in CIECAM97s: (a, b) gamut boundary of CRT and a plane of constant hue angle φ, and (c) their intersection and the intersection of the resulting two-dimensional boundary polygon with a pair of lines. *Source:* J. Morovic, and Luo, M. R., *Color Res. and Appl.*, 25, 394-401, 2000.

1. For each pair of neighboring points in the polygon, calculate the equation of the line determined by them.
2. For each of these $n + 2$ lines, calculate their intersection with l and, if it lies between the two points from the polygon, then it is an LGB point.

Depending on the shape of the gamut boundary, this procedure can result in varying numbers of LGB points for a given line, all of which constitute the LGB along that given line. It is up to the gamut mapping algorithm to deal with cases in which a larger number of LGB points than expected is calculated. Normally, the algorithm results in two points per line — the maximum and minimum along it (e.g., points b_1 and b_2 in Figure 10.9c).

10.3.4 *The magnitude of gamut mismatch*

Now that we have a means of calculating the gamut boundaries of sets of colors, it is useful to get a clearer idea of how big gamut differences can be between various media, and of how serious they are in our example scenario.

10.3.4.1 *Imaging medium gamuts*

To understand the first issue better, Figure 10.10 shows the color gamuts of a range of media when seen in isolation under a number of levels of ambient illumination, and Figure 10.11 shows their gamut volumes. Note that the change of ambient illumination also affects surround conditions for self-luminous media and thereby also alters the state of a viewer's adaptation.

The gamuts shown in Figure 10.10 and 10.11 are those of a set of media viewed under a range of levels of ambient illumination whereby these are designated by the luminance of a near-perfect diffuser as measured under those conditions. The observer from whose point of view this is computed is the CIE standard colorimetric observer,[7] and the color space is the *Jab* space

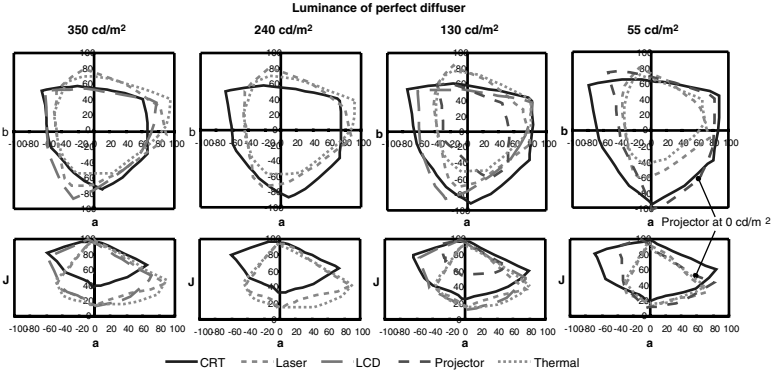


Figure 10.10 Medium gamut boundaries.

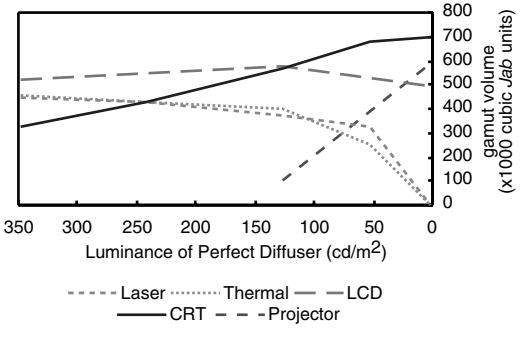


Figure 10.11 Medium gamut volumes.

of the CAM97s2 model. For details of this how these data were obtained, see Morovic et al.[50] and also note that this color appearance model is virtually the same as CIECAM97s, so these figures can be directly compared with the other gamut plots of this chapter. Furthermore, the first row of plots in Figure 10.10 are projections of the gamuts onto the *ab* plane, and the second row of plots are the intersections of the plane having a constant *b* value of zero with the gamut boundary, obtained using the FSLGB method discussed above. Overall, it can be seen from these gamuts that they vary dramatically and that gamut mapping is necessary for reproducing images between any pair of them.

10.3.4.2 Gamut mismatch in example scenario

The differences between the original and reproduction medium gamuts of our scenario are readily visible from Figure 10.8. Figure 10.12 shows the difference between the original image and reproduction medium gamuts more clearly. Note that the original medium (the CRT display) is seen in a dark room alongside the printed reproduction medium, which is viewed in a viewing booth that has a level of illumination such that the luminances of the white points of the two media are similar.

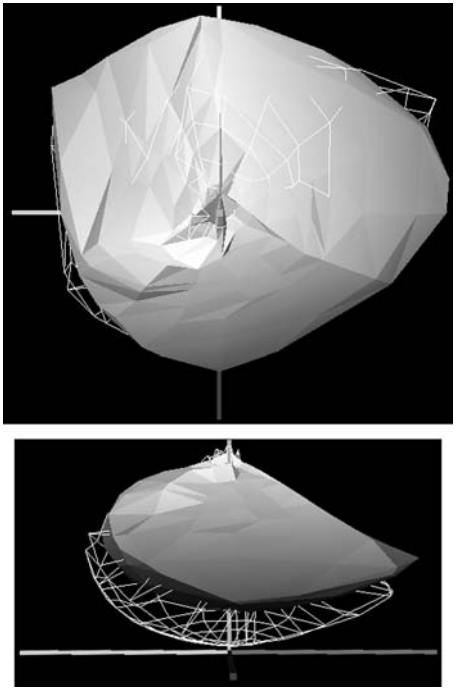


Figure 10.12 Gamuts of the original image (mesh) and the reproduction medium (solid color).

It can be seen from Figure 10.12 that significant parts of the original image's gamut are in parts of color space that are unachievable by the reproduction medium. It is also useful to understand what proportion of the image's pixels are affected by the gamut mismatch, and, in the case of our scenario, it is 78.8% (Figure 10.13). In other words, more than three quarters of the original image's pixels have appearances that cannot be reproduced by the color laser printer in our scenario. The use of a GMA is therefore essential.

This extent of gamut mismatch was determined using the GBD and LGB methods discussed above, by having the GBD for the reproduction medium and then calculating LGBs for lines determined by a center E with Jab values of [50, 0, 0] and each of the image's pixels in turn. A pixel was then judged to be out of gamut if it was farther from E than the reproduction gamut boundary along the line determined by that pixel.

10.4 Gamut mapping algorithms

Now that we have (a) covered the preliminary issues of rendering intent and gamut mapping color spaces, (b) had a careful look at calculating the gamuts of images as well as imaging media, and (c) obtained a clear idea of how serious gamut mismatches can be, we can finally proceed to discussing gamut mapping itself. The reason why gamut mapping can be done only after these issues (and the issues of medium/device characterization modeling and color appearance modeling, which are covered elsewhere) are satisfactorily resolved, is because it is at the heart of the cross-media color reproduction chain and is impossible to look at in isolation. This means both that significant effort needs to be invested in solving other tasks before one can turn to gamut mapping as well as that gamut mapping cannot be evaluated on its own—it is invariably part of a larger system.

Figure 10.13 Out-of-gamut pixels in original image (shown as gray).

This is also why the example scenario was introduced in this chapter, as it will allow a more concrete look at various gamut mapping solutions. However, it also means that the observations made here about their performance might not apply in other scenarios. This is because, whenever our original image is reproduced between other media and/or when other images are used, the context in which a gamut mapping algorithm (GMA) is executed changes. Because of this, the present section will include both a detailed account of how a number of algorithms perform in our scenario and a more general discussion of what might happen under other conditions.

Before proceeding, a more rigorous definition of gamut mapping will be provided. In this chapter, gamut mapping will be understood as "a method for assigning colors from the reproduction medium to colors from the original medium or image (i.e., a mapping in color space)."[46] In other words, after an image has passed through the medium characterization and appearance modeling stages, and color space values are available for each of its pixels, a GMA is executed so as to ensure that each of the color space values in an original image end up being achievable in the reproduction medium; i.e., that they end up inside its gamut. This means that gamut mapping algorithms have the color space values representing an original image as the inputs, and they result in corresponding color space coordinates for the reproduction image that are inside the reproduction medium's gamut boundary.

The remainder of this chapter describes how to get from these inputs to outputs that satisfy the chosen rendering intent. In line with the vast majority of existing research, the algorithms discussed in the following sections transform image pixels on a one-by-one basis, whereby the transformation depends only on the pixel's color. Furthermore, GMAs will be assessed from the point of view of the accurate rendering intent. Departures from this general trend will be dealt with later. Notice also that the intent is to present some basic algorithms in sufficient detail for the reader to be able to implement them and to describe more advanced approaches only briefly. The reader can then follow up in the references relating to these algorithms to obtain more details.

10.4.1 Gamut clipping

The most obvious and simplest kind of approach is to modify only those colors in an original image that are outside the reproduction gamut and to leave those that are already inside it untouched. This type of approach is referred to as *gamut clipping*, and there are a number of ways of doing it.

10.4.1.1 Minimum ΔE clipping algorithms

First, one can perform this mapping so that each original color is transformed to that color in the reproduction gamut that has the smallest color difference from it. This kind of gamut clipping is referred to as *minimum ΔE*, as it minimizes the color-by-color differences between original and reproduction

images with the intention of accurately reproducing the original. As this algorithm intends to minimize color differences, there are a number of flavours of it for all the various equations that can be used for quantifying color difference. The simplest of these are Euclidean distance metrics computed in various color spaces (e.g., ΔE^*_{ab} in CIELAB, ΔE_{Jab} in CIECAM97s). There are also advanced color difference formulæ such as ΔE_{CMC},[9] ΔE_{BFD},[30] and ΔE_{94},[8] which are weighted versions of color difference formulæ, and approaches where changes in some dimensions are prevented altogether (e.g., minimum ΔE clipping that keeps the hue predictors of original colors unchanged). Weighted color difference formulae give different weights to differences in the individual dimensions of a color space.

Katoh and Ito[26] have suggested that out-of-gamut colors should be clipped to colors on the reproduction gamut boundary that have the smallest ΔE value calculated using the following weighted color difference formula in CIELAB:

$$\Delta E = \sqrt{\left(\frac{\Delta L^*}{Kl}\right)^2 + \left(\frac{\Delta C^*}{Kc}\right)^2 + \left(\frac{\Delta H^*}{Kh}\right)^2} \tag{10.4}$$

where ΔL^*, ΔC^*, and ΔH^* are differences in lightness, chroma, and hue predictors, respectively, and Kl, Kc, and Kh are the corresponding weighting coefficients. Based on a psychophysical experiment, the authors found that the most accurate reproductions were obtained when the (Kl:Kc:Kh) coefficients were set to (1:2:1) or (1:2:2), which indicates that larger changes are acceptable in chroma than in hue and that the smallest change is tolerated in lightness. These results are also supported by the work of Ebner and Fairchild[10] and Wei et al.[59] The authors have since published the results of an extensive study investigating the suitability of different color difference formulae (Katoh et al.[27]) for the use in minimum ΔE clipping and have found that ΔE_{94} in CIELUV[7] and ΔE_{BFD} in CIELAB give the best results.

Minimum ΔE clipping algorithms are applied such that each of an image's pixels is input to it and treated in the following way:

- The pixel's color is evaluated in terms of whether it is inside the reproduction gamut, for example, using the technique described in Section 10.3.4.2.
  a. If it is inside the reproduction gamut, then the output of the GMA will be identical to its input.
  b. Otherwise, the color with the smallest ΔE is found on the reproduction gamut boundary, and it becomes the output of the GMA.

All that differs between various flavors of this kind of algorithm are the ΔE formulæ used and how the minimum ΔE color is found. Let us next look at how various minimum ΔE gamut-clipping algorithms could be implemented and what effect they have on the image in our example scenario.

10.4.1.1.1 Minimum Euclidean distance gamut clipping. Using this approach, an out-of-gamut color is replaced by the color that has the smallest Euclidean distance to it on the reproduction gamut boundary. One way of implementing this algorithm, if the segment maxima GBD method is used, is to do the following for each given color:[49]

1. For all the triangles whose vertices are neighboring points from the GBD matrix (for a definition of neighboring points, see Section 10.3.2.2.2), do the following:
   a. Calculate the plane determined by the triangle and also calculate the normal of the plane.
   b. Calculate the intersection of the plane and the line determined by the plane's normal and the given point.
   c. If the intersection is inside the triangle, consider it as a candidate output for the algorithm.

2. For all the pairs of GBD points that are from either horizontally or vertically neighboring segments, do the following:
   a. Calculate the line determined by the pair of points, and also calculate a pair of vectors specifying a plane orthogonal to the line.
   b. Calculate the intersection of the line and the plane determined by the vectors from 2a and the given color.
   c. If the intersection is from the line segment determined by the pair of GBD points, consider it as a candidate output.

3. Calculate the Euclidean distance between the given color and all the GBD points.
4. From among the candidates from 1c and 2c and all the points considered in 3, choose the one that has the smallest Euclidean distance from the given color.

Looking at the above algorithm, it can be seen that various sets of original colors will be mapped onto single reproduction colors. More specifically, a given reproduction color from step 1 would be chosen for all the original colors that are along the line that is orthogonal to the plane of the triangle in which that reproduction color is located. Hence, a one-dimensional region of color space is mapped onto a zero-dimensional one. Reproduction color candidates considered in step 2 each correspond to two-dimensional regions of the original gamut, and three-dimensional parts of the original gamut are mapped onto the colors from step 3. What also follows from this is that different parts of color space will be subjected to different degrees of loss of variation (Figure 10.14).

To see the effect of this algorithm when applied in CIECAM97s or CIELAB, see Figure 10.15. Before making any comments about the performance of the algorithm, it is important to be aware of the fact that it is by

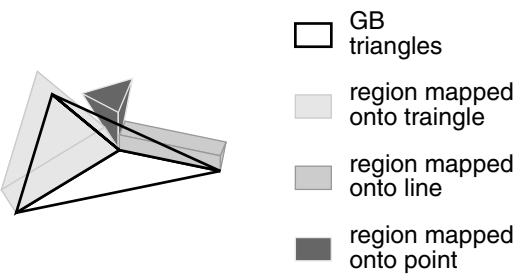


Figure 10.14 Types of variation loss due to minimum ΔE gamut clipping.

(a) (b)

Figure 10.15 Effect of minimum ΔE gamut clipping: (a) clipping in CIECAM97s and (b) clipping in CIELAB. Note that not all aspects of the gamut mapped images can be seen in their present printed form, but their digital versions can be accessed at http://color.derby.ac.uk/~jan/gm/.

definition impossible to maintain all the properties of the original image in the reproduction medium precisely, due to the more limited range of colors available there. Hence, for example, the lightness range will necessarily be more limited, as will the range of chromas and the maximum possible contrast. As a result, it is also impossible to judge whether a GMA has or has not overcome gamut differences well, as the weaknesses in the reproduction could be due either to the GMA's inadequacy or to the limits of the reproduction medium. Making a judgment about the GMA becomes easier when reproductions using other algorithms are available, as it will then be possible to see whether a particular problem is exhibited by only one of the GMA's reproductions (and hence is a limitation of that GMA) or by all reproductions (in which case it could, but does not have to, be due to the limitations of the medium).

Given the above caveats, we can now turn to the gamut-clipped reproduction (Figure 10.15) and see what its characteristics are. What can be seen here is that detail in some parts of the original is not preserved in the

reproduction (e.g., the variation in the grass as well as the dark leaves is lost or at least reduced), that some parts of the image change significantly (e.g., the blue top), and that the reproductions have an overall reduction in contrast.

10.4.1.1.2 Hue-preserving minimum ΔE gamut clipping. The hue-preserving version of minimum ΔE clipping is a frequently used alternative to the previous algorithm. It has the advantage of maintaining all of the original pixels' hue angle predictor values, which in some cases can be considered more important than the absolute minimization of color difference. One way of implementing this algorithm is to use Euclidean distance in planes of constant CIECAM97s h and to process an original image by carrying out the following for each pixel in turn, as described below:

1. Calculate the intersection between the reproduction gamut boundary and the plane of constant hue angle having the given pixel's hue using the FSLGB algorithm.
2. For each of the $n + 2$ line segments of the two-dimensional reproduction gamut boundary, do the following:
   a. Compute the line segment's normal and the line l determined by it and the given color (these lines are shown as dashed gray lines in Figure 10.16).
   b. Compute the intersection between l and the line segment (shown as black triangle outlines in Figure 10.16).
   c. If the intersection is within the line segment, compute the distance between the intersection and the given color, as that intersection is a candidate output for the GMA (shown as white triangles with gray borders in Figure 10.16).
3. For each of the $n + 2$ vertices of the two-dimensional reproduction gamut boundary, calculate the distances between it and the given color. The vertex with the smallest distance is a candidate output value.

(a) (b)

Figure 10.16 Schematic view of hue-preserving minimum ΔE gamut clipping.

4. The final output becomes the candidate from steps 2 and 3 that has the smallest distance from the given color.

The reason why we have to perform step 3 regardless of whether step two results in any candidates is shown in Figure 10.16b. There is a candidate after step 2, but it is not the nearest color on the reproduction gamut. As can be seen from Figure 10.16, whole areas of color space are mapped onto either GB line segments or GB vertices. Using this algorithm, the result of reproducing the original in our scenario is shown in Figure 10.17.

The most significant limitation of this algorithm is that it results in the variation that was present in some parts of the original being lost in the reproduction. Enlarged examples are shown in Figure 10.17b. Notice that the parts of the image where this problem arises are the same as those that, in Figure 10.13, were shown to be outside the reproduction gamut. The reason for this artefact, which all gamut clipping algorithms share in different ways, is that gamut clipping results in whole subvolumes of color space from the original being mapped onto surfaces in the reproduction's color space. This also implies that colors along certain lines through color space from the original are mapped onto a single color in the reproduction; therefore, their differences are lost.

Before moving onto the next algorithm, it is also useful to understand the key role that the gamut mapping color space plays. To help with this, Figure 10.18 shows the result of using CIELAB (Figure 10.18b) instead of CIECAM97s (Figure 10.18a). There, it can be seen that, while the in-gamut parts of the image are the same (as expected), the other areas are reproduced differently when mapping in the two color spaces. For example, the CIELAB reproduction preserves more variation in the blue area, while the CIECAM97s image shows more detail in green and low-lightness regions.

10.4.1.2 Other clipping algorithms

In terms of algorithms that intend to use solely gamut clipping, no other strategies have been tried apart from the two types of approaches discussed

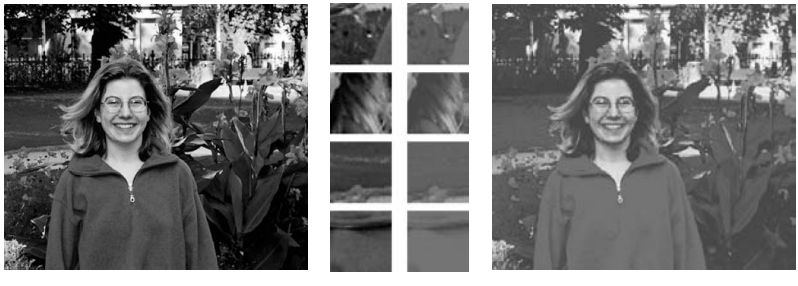


Figure 10.17 ***(See color insert)*** Using hue-preserving minimum ΔE gamut clipping: (a) original image, (b) enlarged details, and (c) gamut-mapped reproduction.

(a) (b)

Figure 10.18 Hue-preserving minimum ΔE gamut clipping: (a) image gamut mapped in CIECAM97s and (b) image gamut-mapped in CIELAB.

above, i.e., minimum ΔE or hue-preserving minimum ΔE. The reason for there being no lightness-preserving or chroma-preserving clipping algorithms is that imposing a constant lightness or chroma constraint would, in many cases, result in no solution being found. Doing this for hue, however, is possible, as all color reproduction media in ordinary use allow for some colors to be reproduced for every hue.

As an alternative to constrained minimum ΔE clipping, another approach would be to clip colors along other kinds of vectors that, from a given original color, point toward a point on the reproduction gamut boundary. This could be done by having all the vectors point toward the center of the reproduction gamut or toward a different point, depending on the original color. To illustrate this kind of approach, Figure 10.19 shows the result of clipping along lines that have constant hue angle and go toward a center of gravity (E) with Jab = [50, 0, 0]. Comparing this reproduction with Figure 10.18a shows that this kind of clipping preserves more variation but at the expense of making the colors in the image more different from the original than was the case with hue-preserving minimum ΔE. Hence, it can be seen that there are conflicting demands made on a GMA when we want both individual colors to be close to the original and also want to preserve differences between colors (and therefore detail).

If we simply consider a monochromatic (one-dimensional) case in which we have two original lightness values of $J_1 = 10$ and $J_2 = 15$, and we have to reproduce them on a medium with a lightness range starting at 20, then we have two kinds of options. The first is to reproduce both $J_1' = J_2' = 20$, whereby we have the smallest difference on a color-by-color basis. The second is reproducing J_1' as 20 and making J_2' greater than 20 (for example, 25). In this manner, we preserve the difference exactly. In essence, all gamut mapping algorithms try to find the best balance between these two poles of individual accuracy and preservation of differences. This section has addressed the first

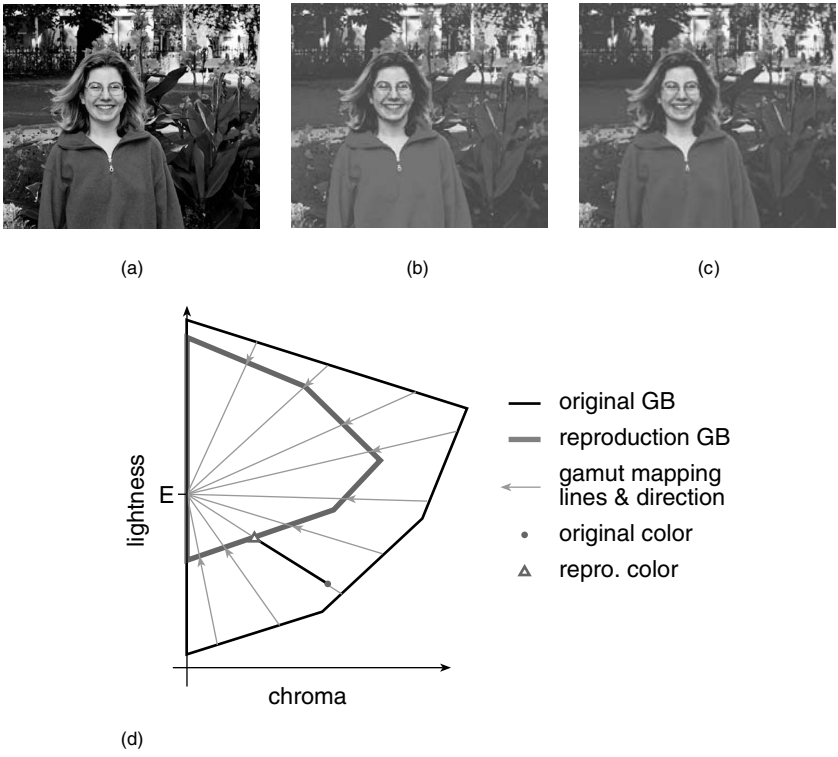


Figure 10.19 (a) Hue-preserving minimum ΔE clipping, (b) original, (c) clipping toward *Jab* = [50, 0, 0], and (d) schematic view of clipping toward *E*.

of these approaches. The following section provides information about the second.

10.4.2 Simple gamut-compression algorithms

As indicated above, gamut-compression algorithms are primarily focused on preserving variation, and they do this by being applied to all original colors rather than only the out-of-gamut ones (as was the case with gamut clipping). In general, gamut-compression algorithms work by changing the position of a color along specific lines. Before the gamut mapping, some colors along that line are out of gamut; after the gamut mapping, all colors end up being inside the reproduction gamut (Figure 10.20).

Algorithms then either compress individual dimensions of a color space separately (i.e., they are *sequential* in that they deal with lightness first and then with chroma); or compress along lines that alter more than one dimension (i.e., they are simultaneous). In addition to considering the lines along which compression is carried out, the center of gravity toward which colors are compressed and the type of compression employed need to be decided

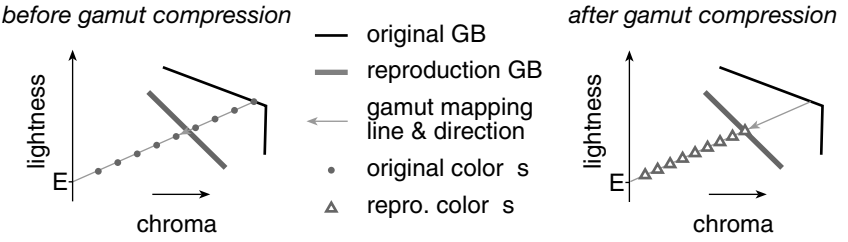


Figure 10.20 Gamut compression along a line toward the center of gravity (E).

as well. In terms of compression types, the most important ones are linear, piece-wise linear, and various kinds of nonlinear mapping. Also note that the vast majority of compression algorithms preserve a color's hue and, unless specified otherwise, this will also be true for the algorithms introduced in the following sections. In fact, as algorithms are implemented in color spaces, all they can do is preserve a particular space's hue predictor (as was mentioned above).

To compress colors along a given line in a linear way, the following equation can be used:

$$C_R = C_O \times G_R / G_O \tag{10.5}$$

where all four variables are distances along the line from the center of gravity; *E* and *O* and *R* represent the original and reproduction, respectively; and *C* and *G* represent the given color and gamut boundary.

10.4.2.1 Simultaneous compression algorithms

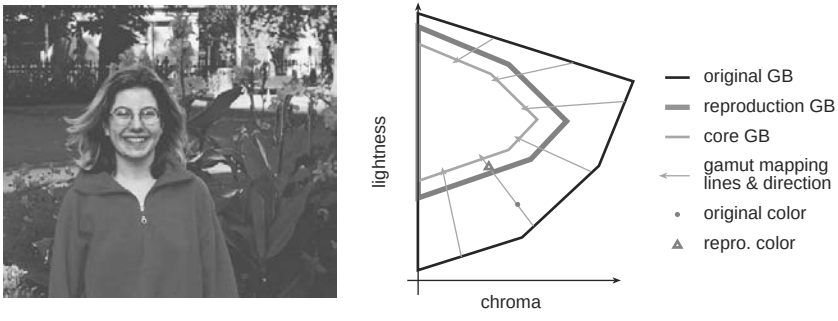
The simplest form of gamut compression consists of specifying a single rule for determining the line along which to map colors and then to apply some kind of compression along that line. For example, linear compression can be carried out according to Equation 10.5 along the lines along which clipping was done in Figure 10.19 (i.e., lines toward a center of gravity (*E*) with *Jab* = [50, 0, 0]). In addition, distances in Equation 10.5 will be distances from *E*. (This algorithm was used previously by Morovic and Luo;[44] in this article, it was referred to as SLIN.) The result of using SLIN can be seen in Figure 10.21a, and Figure 10.21b shows the result of changing *E* in a hue angle dependent way whereby it is set to the point on the lightness axis having the lightness of the cusp at that hue angle. (The cusp at a given hue angle is the color with the largest chroma.) This second approach will be referred to as CUSP, and it is a mapping direction that is used in some of the composite GMAs that will be introduced later. A point to note from Figure 10.21 is also that changes to the algorithm sometimes have a very small impact on the result, as both the reproductions obtained from SLIN and CUSP look very similar there. One needs, however, to bear in mind that the algorithms

(a) (b)

Figure 10.21 (a) SLIN reproduction and (b) CUSP reproduction.

discussed in this chapter are illustrated using a single image reproduced between a single pair of media and that the performance of various algorithms in this scenario cannot be taken as an indication of how they will perform in other contexts.

Another algorithm that has a simple structure and that belongs to this category is TOPO, which has recently been introduced by MacDonald et al.[33] It works by first setting up a core gamut boundary, which is a compressed version of the reproduction boundary and which delimits a region in color space that will be left unaltered. Then, lines of compression are determined in planes of constant hue angle, based on linking points on the original and core boundaries that have the same relative distance along the boundary. Colors are then mapped along these lines in a piece-wise linear way so that there is no change in the core region, and then linear compression occurs between the original and reproduction gamuts outside it. The result of applying TOPO in the scenario, as well as a schematic view of it, can be seen in Figure 10.22. Further details of it can be found in Reference 33.



(a) (b)

Figure 10.22 (a) Reproduction obtained using TOPO and (b) schematic view of TOPO.

In terms of other GMAs based on a single guiding principle, the approach proposed by Katoh and Ito[54] is of particular interest, as it sets forth a way of performing gamut compression along lines determined by smallest color difference.

10.4.2.2 Sequential compression algorithms

An alternative to gamut compression is to deal with the dimensions of a color space one by one. The most common strategy here is to leave hue angle unchanged and then to compress lightness before chroma.[24] Such a GMA first compresses the lightness of a color based on the lightness ranges of the original and reproduction gamuts and then compresses chroma based on the chroma ranges of the two gamuts at the color's compressed lightness. To implement this approach, lightness can be compressed in the following way so as to map the original range onto the reproduction range:

$$J_R = J_{Rmin} + (J_O - J_{Omin}) \times (J_{Rmax} - J_{Rmin})/(J_{Omax} - J_{Omin}) \tag{10.6}$$

where J is CIECAM97s' lightness predictor, *min* and *max* are the minimum and maximum J of a gamut, and O and R refer to original and reproduction, respectively. Chroma could then be compressed using Equation 10.5 whereby the center of gravity for a particular color would be the point on the lightness axis with the same lightness as that color. As a result, the distances in Equation 10.5 would simply be the chroma values of the four points referred to there. To then gamut map a color with this algorithm, the following procedure could be used (this algorithm is referred to as LLIN in Reference 44):

1. Compute original and reproduction gamut boundaries.
2. Compress the original gamut boundary according to Equation 10.6.
3. Compress the given color's lightness according to Equation 10.6.
4. Compress the given color's chroma according to Equation 10.5, and use the result of step 2 as the original gamut boundary.

The result of using this algorithm is shown in Figure 10.23 and, as can be seen, this reproduction does indeed preserve more of the variation of the original than was the case for the GMAs examined so far. However, it can also be seen that the pixel-by-pixel differences between original and reproduction are now greater (e.g., the flowers are less chromatic than before, and even parts of the image that were in-gamut to begin with have been changed).

10.4.2.3 Choosing the original gamut

Unlike gamut clipping, which uses only information about the original color and the reproduction gamut boundary, gamut-compression algorithms also change their behavior on the basis of the original gamut boundary, from

(a) (b)

Figure 10.23 (a) Original and (b) LLIN compressed reproduction.

which they determine the degree of compression that is applied. As a result, an additional question needs to be answered when using a gamut compression algorithms — that of whether to use the original image gamut boundary or the original medium gamut boundary.

The argument for using the original image boundary is that, in this way, compression will be done only to the extent required by a given image, and it has been shown in a number of studies[14,39,53,59] that this does indeed result in more accurate reproductions. Using medium gamuts, on the other hand, has the advantage of allowing for the transformation between a pair of media to be computed only once and then applied to a large number of images. Furthermore, it will result in a given original color always being reproduced by the same color in the reproduction medium. This can be of importance if an important object is present in a number of images, as using the original medium gamut will ensure that it will always be reproduced in the same way.

In terms of our scenario, the difference between the two gamuts is shown in Figure 10.24, and it can be seen there that, while the difference is not significant in some parts of color space, it is dramatic in others. The effect of using the image gamut boundary instead of the original gamut boundary (which is what was used for all the previous examples) can be seen in Figure 10.25. What is most noticeable is that many parts of the image are more colorful when the image gamut boundary is used. This is because, when compressing according to the medium gamut, the image was reduced in chroma so as to accommodate all the possible colors from the original medium.

10.4.3 Composite gamut mapping algorithms

While the GMAs discussed in Sections 10.4.1 and 10.4.2 were all simple in principle, and sufficient detail was provided for their implementation, the gamut mapping solutions discussed here are more complex, and the reader

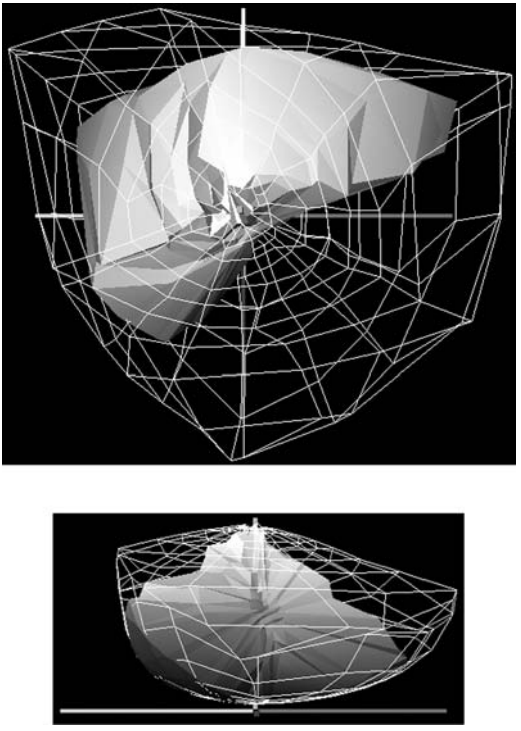


Figure 10.24 Original image gamut (solid) and original medium gamut (mesh).

(a) (b)

Figure 10.25 LLIN reproductions using (a) the original medium gamut or (b) the original image gamut.

will be referred to their original authors for detail. The reason behind the development of these solutions is that, among the simple algorithms described in the previous sections (with the exception of TOPO, which is a recent addition to them), some performed well in one part of color space and others in another. As a result, attempts have been made to combine these different kinds of behavior so as to produce a solution that works well universally.

An early attempt to do this is the GCUSP algorithm proposed by Morovic and Luo,[43] which was developed following an experimental evaluation of some of the simple approaches covered in the previous section (Figure 10.26a). From there, it was found that the CUSP algorithm worked best for colors with large chroma values, whereas the LLIN algorithm performed best for near-neutral colors. As a result, GCUSP consists of a chroma-dependent lightness compression followed by the CUSP algorithm. The chroma-dependent lightness compression results in full linear lightness compression

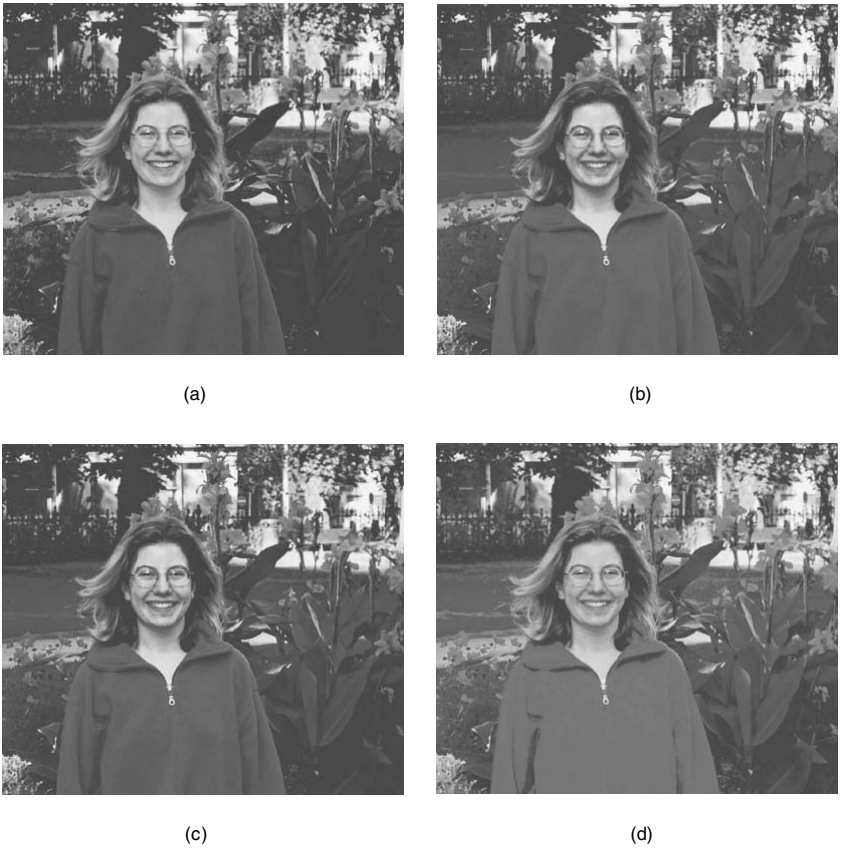


Figure 10.26 ***(See color insert)*** Reproductions made with (a) CGUSP, (b) sigmoidal GMA, (c) SGCK, and (d) CARISMA algorithms.

being applied to achromatic colors, virtually no compression being applied to colors with chromas above around 70 units, and the level of compression changing smoothly between these. Having this kind of chroma-dependent compression means that, around the neutral axis, there is LLIN-type behavior whereas, at high chromas, GCUSP gives results like the CUSP algorithm. In addition to this being done on an experimental basis, there is further justification for this approach in preserving more lightness for colors that have primarily lightness information, and preserving more chroma for colors where chromatic content is significant. For implementation details, see Morovic and Luo.[44]

The algorithm developed by Braun and Fairchild[3] also belongs to this category, and it consists of a sigmoidal lightness compression followed by piece-wise linear compression toward the lightness of the cusp on the lightness axis (Figure 10.26b). Furthermore, the authors of this algorithm also introduce a way of making the shape of the lightness compressing function dependent on the lightness histogram of a given original image so as to preserve more variation in the part of the lightness range, where most of an image's colors are. For details on implementing this algorithm, see Reference 3.

The two above-mentioned algorithms were then combined by the CIE's Technical Committee 8-03 on Gamut Mapping into a single solution that is then recommended as a reference algorithm as part of the TC's guidelines for evaluating gamut mapping algorithms.[42] This algorithm (SGCK) consists of chroma-dependent sigmoidal lightness compression followed by piece-wise linear compression toward the lightness of the cusp on the lightness axis (Figure 10.26c). Details of its implementation can be found in Morovic.[41]

In addition to these GMAs, a number of other solutions belong to this category and are of interest. Among these, the CARISMA algorithm is particularly important, as it is based on analyzing the results of how professional scanner operators reproduce images from transparency to print. The algorithm starts with overall linear lightness compression, changes hue based on the primary hues of the original and reproduction media, and, finally, performs compression along different lines of constant hue angle depending on the different relationships between the two gamuts at various hues (Figure 10.26d). For details of implementation see Morovic and Luo.[44] Other algorithms of interest are the ones proposed by Herzog and Büring,[16] Braun et al.,[4] and Motomura.[51]

10.4.4 Other algorithms for mapping into a smaller gamut

In addition to the gamut mapping approaches discussed in the previous sections, there are also solutions that either are intended to improve gamut mapping performance for certain types of images or take into account spatial relationships in the original image. These GMAs will be discussed in the following sections.

10.4.4.1 Image-type specific GMAs

While the vast majority of gamut mapping research is aimed at developing solutions that can be used for images of any kind, some work has also been aimed at developing better solutions for specific types of images. The most notable effort here is the work by Braun et al.,[5] who have developed a solution specifically for the reproduction of business graphics. Having recognized the fact that, in reproducing business graphics, a key issue is the "clean" reproduction of primary colors, they have proposed a GMA that ensures the mapping of the original medium's primaries onto those of the reproduction medium and thereby providing a solution tailored to those who need to reproduce this kind of image.

10.4.4.2 Spatial GMAs

As mentioned at the beginning of Section 10.4, the vast majority of GMAs treat each color in isolation, and a given original color is therefore always reproduced in the same way, regardless of what image it is in and of what colors surround it in the original. There are, however, a number of proposed solutions that take into account spatial factors and that will result in a given color being treated differently, depending on its surrounding colors in an image.

The earliest one of these is described in the paper by Meyer and Barth,[38] where lightness compression is done in a manner that takes into account spatial characteristics. To this end, lightness at a given point $[x, y]$ in the image is expressed as $F(x, y) = I(x, y) \times R(x, y)$, where I is the spatially slowly varying contents in an image (depending on the illuminant), and R is a spatially rapidly varying function containing the image detail itself. As it is, the I component, which controls the lightness range, is compressed similarly to the method of Equation 10.6. Chroma is then dealt with in a conventional way.

Nakauchi et al.[52] also propose an approach that, in effect, takes spatial properties into account as they look at gamut mapping as an optimization problem of finding an image that satisfies the following criteria:

1. It is perceptually closest to the original.
2. All of its pixels are within the reproduction gamut.

The perceptual difference between the reproduction and the original is then defined by applying bandpass filters to the images and obtaining their difference in a way that takes into account human contrast sensitivity. A method for calculating an image that minimizes perceptual difference and remains in gamut is given in the paper.

Another approach, taken by McCann,[37] uses the Retinex theory[28] to propose gamut mapping that aims to preserve the ratios between the colors that are present in an original. Finally, the most recent of these methods is the one proposed by Balasubramanian et al.,[1] which aims to preserve spatial

luminance or lightness variation in a way similar to Meyer and Barth's method.

10.4.5 Gamut expansion algorithms

Little work has been published in this area to date, as the issue involves considerable complexity in terms of specifying both what the mapping should take as parameters and what the resulting image's ideal characteristics should be. To some extent, the development of gamut expansion algorithms is also linked with a better understanding of preferred color reproduction, which itself has not been researched a great deal in recent times. The kind of applications that would benefit from gamut expansion are those in which images from more limited media are rendered on media with larger gamuts. For example, printed images rendered on a television display could benefit from making use of parts of color space that were not available to the original, and a digital camera image taken under lower levels of illumination could be enhanced by enlarging its gamut. The principal difficulty here is that, while some scenes might benefit from gamut expansion, others might deteriorate. An example of where gamut expansion might produce negative results is for images with skin tones, which after expansion might appear too colorful. Objects that include pastel colors could also suffer from this kind of change. To find out more about the work done so far on gamut expansion, refer to Hoshino[17,18] and Kang et al.[25]

10.5 Factors affecting gamut mapping algorithms

Now that a wide range of gamut mapping algorithms have been introduced and illustrated in the context of our scenario, it is important to discuss once again the fact that all these illustrations only provide information about how a given algorithm performs in our scenario. If a different media pair or different images were used, the result might well be quite different from the ones shown here. Therefore, the next two sections describe the impact media and images can have on the performance of gamut mapping algorithms. This discussion is based on Reference 41.

10.5.1 Media

The nature and magnitude of the differences between an original and a reproduction medium play an important role in determining whether a given algorithm is suitable for mapping images between the pair of media. Where there are relatively small differences between the two media, the method for overcoming them is less important than in a case where the differences are large. This is so because the potential of difference between various gamut mapping solutions is smaller in the former than in the latter cases. To illustrate this, Figure 10.27 shows the result of using hue-preserving minimum ΔE clipping and the LLIN algorithm when mapping the image in our

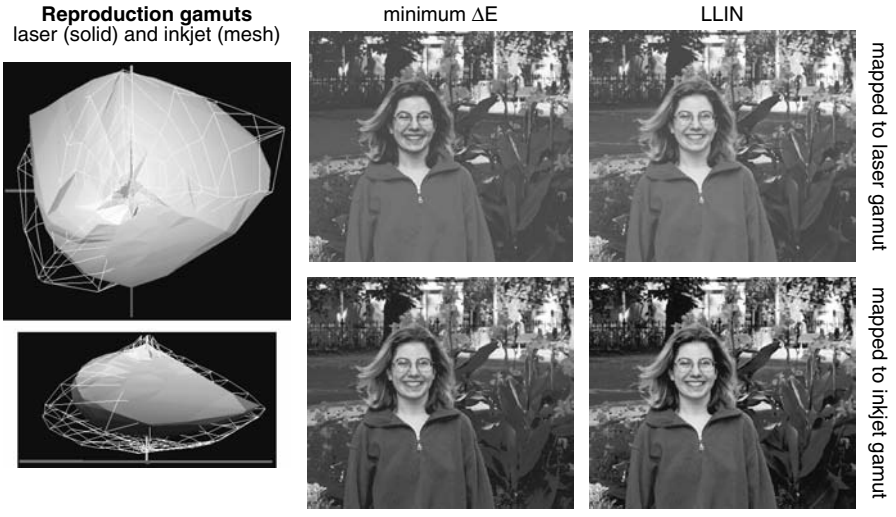


Figure 10.27 Reproducing the original on a small-gamut laser printer and a large-gamut inkjet printer.

scenario, as well as when mapping it to an inkjet printer with a significantly larger gamut than the laser printer of our scenario.

In fact, existing literature seems to support this conjecture, as it shows that, when medium differences are small, clipping algorithms perform better than compression algorithms[39,48] and that the opposite is true for large gamut differences.[43] Hence, when gamut mapping between a pair of printed media, clipping might be the best solution whereas, when mapping between a transparency projected in a dark room and a CRT seen in an office environment, compression might be a better approach.

In addition to the medium differences, another important issue is to ensure that the gamut boundary information used by the GMA represents the viewing conditions for which the mapping is carried out. The importance of calculating GB information under the conditions used for viewing both the original and reproduction was discussed by Morovic et al.,[50] and the following discussion is based on that paper. For example, a CRT display's gamut could be calculated on the basis of colorimetric measurements taken in a dark room, and a print's gamut could be calculated using spectrophotometric measurements relating to the print being viewed under a standard illuminant (e.g., D50). If the system is then used to match images between the CRT and a print under office lighting conditions, then any mismatch would, to a great extent, be caused by the difference between the conditions under which the media were characterized and the conditions under which they are used. To illustrate this point, Figure 10.28 shows the gamuts of a CRT and a print when viewed simultaneously in a room under a range of ambient illumination levels. As can be seen in this figure, the relative shapes

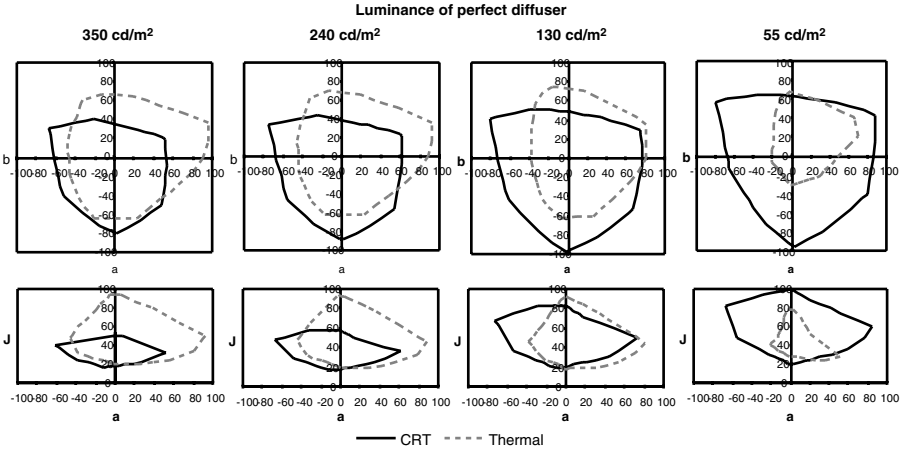


Figure 10.28 Color gamuts of CRT and print under different illumination levels.

of the gamuts of these two media change dramatically with changes of ambient illumination level, and it is therefore crucial to have correct gamut boundary data when performing the gamut mapping. Part of the reason for the opposite direction in the two gamuts' changes is that increasing the level of ambient illumination results in flare being added to the output of the CRT, whereas it boosts the intensity of the printed colors. The effect of flare on displays is also discussed by Sharma.[55]

10.5.2 *Images*

In the same way that the original and reproduction media are important factors influencing GMA performance, the image that is being reproduced plays a key role. While the importance of images has long been known,[3,32,43,58] a systematic study of why one image needs to be reproduced differently from another image has only recently commenced.[47] In previous studies it has been conjectured that it is the image type (in the sense in which an image can be a business graphic, a portrait, or an outdoor scene) that needs to be analyzed to decide how to reproduce it, but this claim is made without substantial evidence. The results of the above-mentioned systematic study show that it is not an image's gamut[48] or its lightness, chroma, or combined lightness and chroma histograms,[56] but the full 3-D image color histogram[57] that influences how an image will be reproduced by different gamut mapping algorithms. As such, image characteristics have been shown to be of importance in cross-media reproduction, and they are a factor to be considered whenever gamut mapping is carried out.

What is most important to bear in mind, in relation to the role of images in gamut mapping, is that a given algorithm is very likely to perform differently for different images. Therefore, whether a GMA will work well for reproducing a particular image depends not only on that GMA or the media

between which the reproduction is to be made, but also on the image itself. To illustrate this, Figure 10.29 shows what happens when three images are reproduced between the media from our scenario using two different GMAs.

What can be seen here is that the two GMAs perform differently for the three images and also that the difference between the GMAs is greater for some images than others (i.e., their difference seems smallest for the middle image and greater for the top and bottom ones). Also noticeable is the fact that, while LLIN performs better for the bottom image, hue-preserving minimum ΔE does better for the top image and that the two are similar for the middle image. Even if the reader disagrees with the judgments about which algorithm does better for which image, it is clear from Figure 10.29 that the difference between the performance of algorithms depends on image as well as medium.

10.6 Summary

This chapter provides an overview of gamut mapping from the point of view of what happens to an image when it is reproduced between a pair of media.

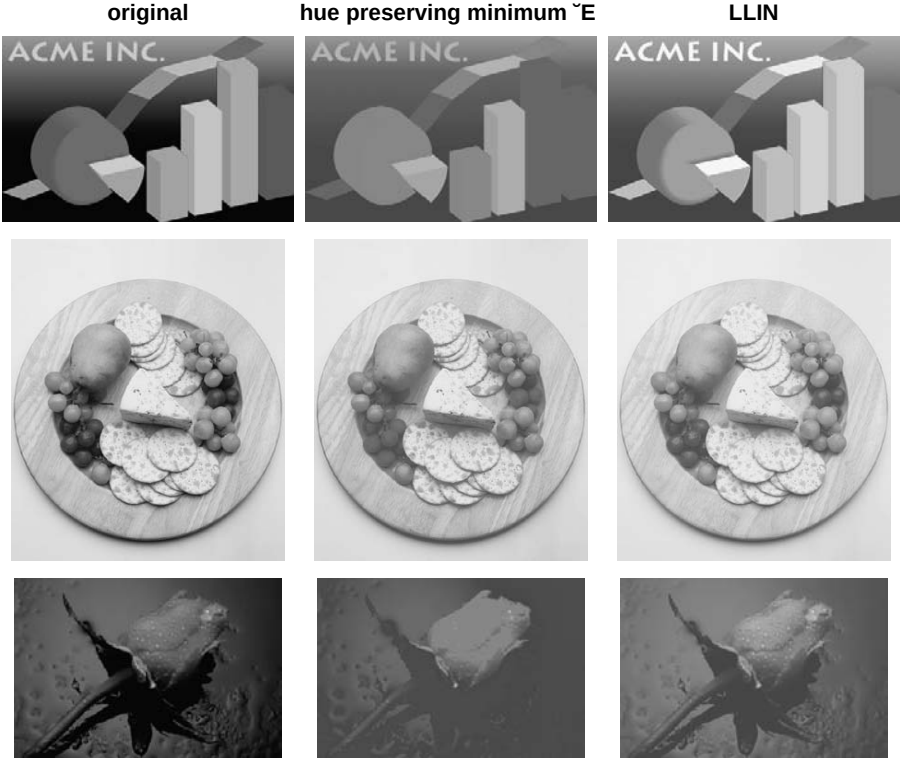


Figure 10.29 ***(See color insert)*** Impact of image characteristics on GMA performance.

Before defining gamut mapping, we introduced issues related to rendering intents, input and output data, and color spaces, and we considered in detail the implications of what a color gamut is. Various factors of calculating color gamut boundary descriptors were then reviewed, a detailed description was given of the segment maxima method, and image-specific issues were addressed. After covering these preliminary areas, we then turned to a discussion of gamut mapping algorithms themselves, wherein the focus was to describe some basic clipping and compression algorithms with sufficient detail for implementation by the reader. Also introduced were some more advanced and complex solutions, and the areas of image type specific and spatial algorithms as well as gamut expansion were briefly mentioned. Finally, we discussed the importance of the media on which images are reproduced as well as the importance of the images themselves. Furthermore, the majority of algorithms discussed in this chapter were also illustrated in the context of a cross-media color reproduction scenario, which was referred to throughout the chapter.

It is also worth noting that no recommendations were made herein as to what algorithm to use. This is because the topic of gamut mapping is one that is still in a stage of rapid development and is one for which we still do not have a thorough understanding of what determines the proper approach. Hence, this chapter intended to illustrate the complexities of the gamut mapping problem and present a number of existing solutions as equals rather than making any hasty recommendations. The reader who is interested in following up on comparisons made between existing gamut mapping algorithms as well as the progress of gamut mapping research in general is advised to refer to existing and future issues of the *Journal of Electronic Imaging*, *Color Research and Application,* and in particular the proceedings of the IS&T/SID Color and Imaging Conference, held annually in Scottsdale, Arizona.

To conclude, I would like to paraphrase Samuel Butler (II) in saying that gamut mapping of all kinds is like painting — a compromise with impossibilities.

Acknowledgments

I would like to thank Prof. M. Ronnier Luo and Mr. Pei-Li Sun for their friendship and all the gamut mapping research we have done together. Furthermore, I would also like to thank my parents, Ján and Juliana; my brother, Peter; my sisters, Monika and Beátka; and my wife, Karen, for their constant support. Above all, though, I would like to thank God, who is Love.

References

1. Balasubramanian, R., de Queiroz, R., Eschbach, R., and Wu, W., Gamut mapping to preserve spatial luminance variations, in *Proc. 8th IS&T/SID Color Imaging Conference*, 2000, Scottsdale AZ, 122–128.

2. Braun, G. J. and Fairchild, M. D., Techniques for gamut surface definition and visualization, in *Proc. 5th IS&T/SID Color Imaging Conf.*, 1997, 147–152.
3. Braun, G. J. and Fairchild M. D., Image lightness rescaling using sigmoidal contrast enhancement functions, *J. Electronic Imaging*, 8(4), 380–393, 1999.
4. Braun, K. M., Balasubramanian R., and Eschbach R., Development and evaluation of six gamut mapping algorithms for pictorial images, in *Proc. 7th IS&T/SID Color Imaging Conf.*, 1999, 144–148.
5. Braun, K. M., Balasubramanian, R., and Harrington, S. J., Gamut mapping techniques for business graphics, in *Proc. 7th IS&T/SID Color Imaging Conf.*, 1999, 149–154.
6. CIE Publication 51, *A Method for Assessing the Quality of Daylight Simulators for Colorimetry,* CIE, 1981.
7. CIE Publication 15.2, *Colorimetry,* 2nd ed., CIE, 1986.
8. CIE Publication 116, *Industrial Colour-Difference Evaluation,* CIE, 1995.
9. Clarke, F. J. J., MacDonald, R., and Rigg B., Modification of the JPC79 Colour-difference formula, *J. Soc. Dyers Colourists*, 100, 117, 1984.
10. Ebner, F. and Fairchild, M. D., Gamut mapping from below: finding the minimum perceptual distances for colors outside the gamut volume, *Color Res. and Appl.*, 22, 402–413, 1997.
11. Ebner, F. and Fairchild, M. D., Finding constant hue surfaces in color space, *SPIE Proc.*, SPIE, Bellingham, WA, 3300, 107–117, 1998.
12. Ebner, F. and Fairchild, M. D., Development and testing of a color space (IPT) with improved hue uniformity, in *Proc. 6th IS&T/SID Color Imaging Conf.*, IS&T, Springfield, VA, 1998, 8–13.
13. Engeldrum, P. G., Computing color gamuts of inkjet printing systems, *SID Proc.*, 27, 25–30, 1986.
14. Gentile, R. S., Walowitt, E., and Allebach J. P., A comparison of techniques for color gamut mismatch compensation, *J. Imaging Techol.*, 16, 176–181, 1990.
15. Herzog, P. G., Further development of the analytical color gamut representations, *SPIE Proc.*, 3300, 118–128, 1998.
16. Herzog, P. G. and Büring, H., Optimizing gamut mapping: lightness and hue adjustments, in *Proc. 7th IS&T/SID Color Imaging Conf.*, IS&T, Springfield, VA, 1999, 160–166.
17. Hoshino, T., A preferred color reproduction method for the HDTV digital still image system, in *IS&T Proc. Symposium on Electronic Photography,* 1991, 27–32.
18. Hoshino, T., Color Estimation Method for Expanding a Color Image for Reproduction in a Different Color Gamut, U.S. Patent No. 5,317,426, 1994.
19. Hung, P. C. and Berns, R. S., Determination of constant hue loci for a CRT gamut and their predictions using color appearance spaces, *Color Res. Appl.*, 20, 285–295, 1995.
20. Hunt, R. W. G., *The Reproduction of Colour in Photography, Printing & Television*, 5th ed., Fountain Press, England, 1995, 225–240.
21. International Color Consortium Specification ICC.1:2001–04 File Format for Color Profiles [revision of ICC.1:1998–09], 2001, see http://www.color.org/.
22. Inui, M., Fast algorithm for computing color gamuts, *Color Res. Appl.*, 18, 341–348, 1993.
23. Ito, M. and Katoh, N., Three-dimensional gamut mapping using various color difference formulæ and color spaces, in *Proc. SPIE and IS&T Electronic Imaging '99 Conf.*, 1999.

24. Johnson, A. J., Perceptual requirements of digital picture processing, paper presented at IARAIGAI symposium (1979) and printed in part in *Printing World*, 6, February, 1980.
25. Kang, B. H., Cho, M. S., Morovic J., and Luo, M. R., Gamut expansion development based on observer experimental data, in *IS&T/SID 9th Color Imaging Conf.,* 2001 (submitted).
26. Katoh, N. and, Ito M., Gamut Mapping for Computer Generated Images (II), in *Proc. 4th IS&T/SID Color Imaging Conf.*, 1996, 126–129.
27. Katoh, N., Ito M., and Ohno, S., Three-dimensional gamut mapping using various color difference formulae and color spaces, *J. Electron. Imag.,* 8(4), 365–379, 1999.
28. Land, E., The retinex, *Am. Scientist*, 52, 247–264, 1964.
29. Luo, M. R. and Hunt, R. W. G., The structure of the CIE 1997 color appearance model (CIECAM97s), *Color Res. Appl.*, 23, 138–146, 1998.
30. Luo, M. R. and Rigg, B., BFD(l:c) colour difference formula, parts 1 and 2, *J. Soc. Dyers Colourists*, 126–132, 1987.
31. MacDonald, L. W., Gamut mapping in perceptual color space, in *Proc. 1st IS&T/SID Color Imaging Conf.*, IS&T, Springfield, VA, 1993, 193–196.
32. MacDonald, L. W. and Morovic, J., Assessing the effects of gamut compression in the reproduction of fine art paintings, in *Proc. IS&T/SID 1995 Color Imaging Conf.: Color Science, Systems and Applications*, 1995, 194–200.
33. MacDonald, L. W., Morovic, J., and Xiao, K., A topographic gamut mapping algorithm based on experimental observer data, in *Proc. IS&T/SID 8th Color Imaging Conf.*, 2000, 311–317.
34. MacDonald, L. W., Morovic, J., and Xiao, K., Evaluating gamut mapping algorithms, *J. Imaging Sci. Technol.,* 2001 (submitted).
35. Mahy, M., Calculation of color gamuts based on the neugebauer model, *Color Res. Appl.*, 22, 365–374, 1997.
36. Marcu, G., Gamut mapping in Munsell constant hue sections, in *Proc. 6th IS&T/SID Color Imaging Conf.*, IS&T, Springfield, VA, 1998, 159–162.
37. McCann, J. J., Color gamut measurements and mapping: the role of color spaces, *SPIE Proc.*, SPIE, Bellingham, WA, 3648, 68–82, 1999.
38. Meyer, J. and Barth, B., Color gamut matching for hard copy, *SID 89 Digest*, SID, San Jose, CA, 1989, 86–89.
39. Montag, E. D. and Fairchild, M. D., Psychophysical evaluation of gamut mapping techniques using simple rendered images and artificial gamut boundaries, *IEEE Trans. Image Proc.*, 6, 977–989, 1997.
40. Morovic, J., To Develop a Universal Gamut Mapping Algorithm, Ph.D. thesis, University of Derby, 1998.
41. Morovic, J., Colour gamut mapping, in *Colour Engineering: Achieving Device Independent Colour*, Green, P. and MacDonald, L. W., Eds., John Wiley & Sons, New York, 2001.
42. Morovic, J., Ed., *Guidelines for the Evaluation of Gamut Mapping Algorithms*, CIE Technical Committee 8-03, 2001, http://www.colour.org/tc8-03/.
43. Morovic, J. and Luo, M. R., Gamut mapping algorithms based on psychophysical experiment, in *Proc. 5th IS&T/SID Color Imaging Conf.*, 1997, 44–49.
44. Morovic, J. and Luo, M. R., Developing algorithms for universal colour gamut mapping, in *Colour Engineering: Vision and Technology,* MacDonald, L. W. and Luo, M. R. Eds., John Wiley & Sons, Chichester, England, 1999, 253–282.

45. Morovic, J. and Luo, M. R., Calculating medium and image gamut boundaries for gamut mapping, *Color Res. Appl.*, 25, 394–401, 2000.
46. Morovic, J. and Luo, M. R., The fundamentals of gamut mapping: a survey, *J. Imaging Sci. Technol.*, 45(3), 283–290, 2001.
47. Morovic, J. and Sun, P. L., Methods for investigating the influence of image characteristics on gamut mapping, in *Proc. IS&T/SID 7th Color Imaging Conf.*, 1999, 138–143.
48. Morovic, J. and Sun, P. L., The influence of image gamuts on cross-media colour image reproduction, in *Proc. IS&T/SID 8th Color Imaging Conf.,* 2000, 324–329.
49. Morovic, J. and Sun, P. L., Non-iterative minimum ΔE gamut clipping, in *Proc. IS&T/SID 9th Color Imaging Conf.*, 2001.
50. Morovic, J., Sun, P. L., and Morovic, P., The gamuts of input and output colour reproduction media, in *Proc. SPIE*, 4300, 114–125, 2001.
51. Motomura H., Gamut mapping using color-categorical weighting method, in *Proc. 8th IS&T/SID Color Imaging Conf.*, IS&T, Springfield, VA, 2000, 318–323.
52. Nakauchi, S., Imamura, M., and Usui, S., Color gamut mapping by optimizing perceptual image quality, in *Proc. 4th IS&T/SID Color Imaging Conf.*, IS&T, Springfield, VA, 1996, 63–67.
53. Pariser, E. G., An investigation of color gamut reduction techniques, in *Proc. IS&T's 2nd Symposium on Electronic Publishing*, IS&T, Springfield, VA, 1991, 105–107.
54. Katoh, N. and Ito, M., Applying non-linear compression to the three-dimensional gamut mapping, in *Proc. 7th IS&T/SID Color Imaging Conf.,* IS&T, Springfield, VA 155–159, 1999.
55. Sharma, G., LCD displays vs. CRTs: color-calibration and gamut considerations, in *Proc. IEEE, Special Issue on Flat Panel Display Technologies*, 40(4), 605–622, 2002.
56. Sun, P. L. and Morovic J., The influence of image histograms on cross-media colour image reproduction, *PICS 2001*, Montrèal, Canada, 2001, 363-367.
57. Sun, P. L. and Morovic, J., 3D histograms in colour image reproduction, *Electronic Imaging,* 2002.
58. Viggiano, J. A. S. and Moroney, N. M., Color reproduction algorithms and intent, in *Proc. 3rd IS&T/SID Color Imaging Conf.*, 1995, 152–154.
59. Wei, R. Y. C., Shyu, M. J., and Sun, P. L., A new gamut mapping approach involving lightness, chroma and hue adjustment, *TAGA Proc.*, TAGA, Rochester, NY, 1997, 685–702.
60. Zeng, H., Gamut mapping in multiple color spaces, in *SPIE Proc.*, SPIE, Bellingham, WA, 3963, 301–306, 1999.
61. Kress, W. and Stevens, M., Derivation of 3-dimensional gamut descriptors for graphic arts output devices, *TAGA Proc.,* 199–214, 1994.
62. Cholewo, T. J. and Love, S., Gamut boundary determination using alpha-shapes, in *Proc. 7th IS&T/SID Color Imaging Conf.,* 200–204, 1999.

chapter eleven

Efficient color transformation implementation

Raja Balasubramanian
R. Victor Klassen
Xerox Innovation Group

Contents

11.1 Introduction

In many instances along the imaging path, images need their colors transformed. For example, images received in a device-dependent color space must be transformed to a device-independent color space. In many cases, these transformations are complex nonlinear functions, thus making it impractical to process large images in real time. A simple solution would be to precompute the transform for all possible digital inputs and store the corresponding outputs in a lookup table (LUT). This approach results in a perfectly accurate transformation (within limits of numerical computation), and the computational processing is relatively simple (i.e., only multidimensional lookups). However, the storage requirements quickly become prohibitive. For example, 8-bit RGB input requires a LUT of size 2^{24} = 16 M; 8-bit CMYK input requires a LUT size of 2^{32} = 4 G entries. For CMYK output, each entry is at least 32 bits, or 4 bytes. If the price of this quantity of memory is not an issue, the time to access it may be, as it is unlikely that the portions of interest will remain in the processor cache.

An alternative is to build a LUT with a sparser node sampling and to use multidimensional interpolation for input points that do not coincide with the LUT nodes. Constructing such a LUT involves first defining a lattice of nodes that partition the input color space into a set of smaller subvolumes. The desired color transformation is evaluated at these nodes, and the resulting output values are stored at the node locations. Algorithms for LUT construction are described in Chapter 5, so we will not elaborate on that aspect here. We will focus instead on LUT interpolation techniques. Given

an input point for which the color transformation is to be evaluated, multidimensional interpolation involves the following steps:

1. Find the subvolume to which the input color belongs.
2. Retrieve the nodes that define the subvolume.
3. Interpolate among the output values at these nodes to obtain the final output color.

The data may need additional processing before and after these three steps. Examples will be given in the following sections.

LUT interpolation involves a three-way trade-off among (a) LUT approximation accuracy, (b) storage and memory requirements, and (c) computational complexity. In the following sections, a number of LUT interpolation techniques are presented and compared in terms of the trade-offs among these three variables. The best technique and trade-off depends on the requirements and constraints of a given application. In the following discussions, we will use two-dimensional and three-dimensional input spaces for illustrative purposes, bearing in mind that extension to more dimensions is usually straightforward.

Sections 11.2 through 11.4 describe the basic LUT transformation and variants thereof. Sections 11.5 through 11.9 describe algorithms for efficiently processing large amounts of image data through a LUT transformation.

11.2 Interpolation on regular lattices

11.2.1 Regular lattice structures

Define l_i to be a set of real-valued levels along the ith color dimension. A regular lattice L^m in m-dimensional color space is defined as the set of all points $\mathbf{x} = [x_1,\ldots,x_m]^t$ whose ith component x_i belongs to the set l_i. Mathematically, the lattice can be expressed as

$$L^m \{\mathbf{x} \in R^m \big| x_i \in l_i, i = 1,\ldots,m\} \quad \text{or} \quad L^m = \prod_{i=1}^{m} l_i \tag{11.1}$$

where the second expression is a Cartesian product. If s_i is the number of levels in l_i, the size of the lattice is the product $s_1 s_2 \ldots s_m$. A regular three-dimensional lattice is shown in Figure 11.1. It partitions the three-dimensional input space into rectangular subvolumes. In the simplest form, the lattice levels l_i are spaced uniformly along the ith dimension, and $s_1 = s_2 = \ldots = s_m = s$, resulting in a lattice size of s^m. However, the lattice size and spacing of levels along each dimension can be varied to suit the characteristics of the function being approximated. Typical tables have from $8 \times 8 \times 8$ to $33 \times 33 \times 33$ nodes, with s_i being either a power of 2 or one greater than a power of 2. Chapter 5 contains experimental data that compares LUT accuracy for different lattice sizes.

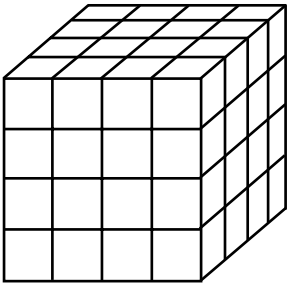


Figure 11.1 A regular lattice in three dimensions.

The main advantage of a regular lattice is that the step of finding the subvolume to which an input color belongs is very simple and accomplished as follows. Assume a color [x, y, z] in a three-dimensional space represented by n bits, and thus values ranging from 0 to $C_{max} = 2^n - 1$. This color is to be transformed using a LUT of size $m \times m \times m$. First, a nearby LUT node is located with indices [X][Y][Z], given by

$$X = \lfloor x\sigma \rfloor,\ Y = \lfloor y\sigma \rfloor,\ Z = \lfloor z\sigma \rfloor; \qquad \sigma = m/C_{max} \tag{11.2}$$

where $\lfloor\ \rfloor$ denotes the rounding down operation. (Throughout this section, C-style indices are used, i.e., the first index of the table runs from 0 to $m - 1$.) If the color [x, y, z] coincides with node [X][Y][Z], the output values stored at this node are simply looked up. Otherwise, the desired output is obtained by interpolating with the seven neighboring nodes, which have one or more of the LUT indices [X][Y][Z] increased by one. These eight nodes form the rectangular subvolume enclosing the input point. The highest numbered nodes (index $m - 1$), when used, never require interpolation. One must either test to avoid interpolating with the next set, ensure that a set of dummy values (which will be multiplied by 0) are allocated beyond the last value, or risk reading beyond the end of allocated memory.

Example. Given an 8-bit, three-separation input space and a $17 \times 17 \times 17$ uniform lattice, we have $\sigma = 17/255 = 1/15$. An input color [23, 36, 190] will locate LUT indices [1][2][12]. The output is obtained by interpolating among the nodes at [1][2][12], [1][2][13], [1][3][12], [1][3][13], [2][2][12], [2][2][13], [2][3][12], and [2][3][13]. Any given input color not on a node [X][Y][Z] is a fraction t, u, and v of the way to the next node in the x, y, and z directions, respectively. The fraction $1/\sigma$ gives the distance between nodes. The fractional distance of x between nodes at X and $X + 1/\sigma$ is $t = x\sigma - X$.

11.2.1.1 Lattice dimensions: power of two or one greater?

There is a long-standing debate over whether the lattice dimensions should be a power of two or one greater. This is largely a question of efficiency. As we shall see, the difference is very small if both are implemented as efficiently as possible. For concreteness, we shall compare 16^3 with 17^3; however, any

such pairing follows the same argument — and one may leave the vicinity of a power of two altogether without paying the imagined penalty.

The key advantage of 16^3 is in addressing the (i, j, k) entry using the (C language) expression

$$base + (((i << 4) + j) << 4) + k,$$

rather than the expression needed for the 17^3 case,

$$base +(((i * 17) + j) * 17) + k,$$

which involves two multiplications, normally slower than the shifts. Thus, once the values of i, j, and k are known, indexing is faster using 16^3. However, these quantities are derived from color separation values, and obtaining them and the interpolation parameters (t, u, and v) is more difficult for a power-of-two LUT.

From the geometry of the table, it clearly contains nodes at 0 and max, with others uniformly spaced in between. Interpolation is between pairs of adjacent nodes, and along any axis, the number of pairs of adjacent nodes is a power of two. If the table size is 17^3, the index of the lower node is found in the high-order bits of the input color separations, and the interpolation parameters are the low-order bits of the same input color separations. Thus, to obtain i and a fixed point value of t, one computes

$$i = x >> 4$$

$$t = x \ \& \ 0xf;$$

whereas, for a table size n that is not a power-of-two-plus-one,

$$i = x / (n - 1),$$

$$t = x - (n - 1)\ i.$$

So it appears that the debate is more about choosing where the multiplications and divisions occur than how many of them there are. As it happens, none of these calculations is necessarily optimal. The number of possible values of x is small (somewhere in the 256 to 4096 range), and so the values of i and t may be obtained by a simple LUT from the value of x, eliminating the division and multiplication in the power-of-two-plus-one case. Note, however, that it is not really the value of i that is needed, but the quantity $n * m * i$, which could just as easily be stored in the lookup table as i. The addressing arithmetic is equivalently

$$base + i * n * m + j * m + k,$$

so the table indexed by x can contain the appropriate values of t as well as the values of $i * n * m$, while the table indexed by y contains the values of u and $j * m$, with the table indexed by z containing k and v.

Thus, both table address calculation and parameter calculation are obtained through a single lookup operation (returning a two-element structure), independent of the table size. With this approach, there is no advantage

to using either a power of two or one greater; any size has the same computational cost. There is one potential advantage to having an odd-sized table, and that is for input color spaces with a neutral-axis aligned dimension (such as L*a*b* or YCrCb), one may arrange to have the neutral axis exactly represented along one column of nodes while maintaining node symmetry about this axis.

11.2.1.2 *Anamorphic lattice structures*

Balasubramanian has proposed using an anamorphic lattice, a variation of the regular lattice structure that offers a more efficient sampling of nodes.[1] The idea is based on the observation that interpolating from an m-dimensional input space to an n-dimensional output space consists of n independent interpolations from m-dimensional input to one-dimensional output. Consider, for example, a LUT transform from display RGB to printer CMY. The output cyan value is calculated by interpolating the cyan values at the surrounding nodes. Similarly, the magenta and yellow outputs are independently calculated by interpolating the nearby magenta and yellow node values, respectively. This being the case, there is no reason to use the same lattice to interpolate the C, M, and Y. For example, one might expect that C has a strong dependence on its complementary input R and a weak dependence on G and B. Likewise, M may show stronger dependence on input G than on R and B. Hence, a suitable lattice for interpolating C would have a dense sampling along R and coarse sampling along G and B. Equivalently, the M signal requires a finer sampling along G than along R and B. The idea is demonstrated in Figure 11.2 for a 2-D transform from R-G to C-M. Experimental results have shown that the average ΔE^*_{94} LUT error is reduced from 2.1 using a standard lattice to 1.1 using an anamorphic lattice of the same size. Thus, an anamorphic lattice improves the size-accuracy trade-off. Note, however, that the anamorphic lattice requires finding the enclosing subvolume for each output separation, thus incurring a small computational overhead.

11.2.2 *Two-dimensional interpolation geometries*

Given a rectangular subvolume and its vertex nodes that surround the input point, the interpolation geometry specifies which subset of those nodes is used to calculate the final color. The two-dimensional geometries of bilinear and triangular interpolation will be described first, as these are used as building blocks for the three-dimensional geometries commonly used in color transformations.

Bilinear interpolation is the natural extension of linear interpolation to two dimensions. It is based on the assumption that the function being approximated is "close enough" to linear between two nodes. That is, it has constant derivative between those nodes. If the values at nodes n_0 and n_1 are p_0 and p_1, respectively, then the value at a location that is a fraction t of the way from n_0 to n_1 is given by $p_0 + (p_1 - p_0)t$. To compute an interpolant a fraction t of the

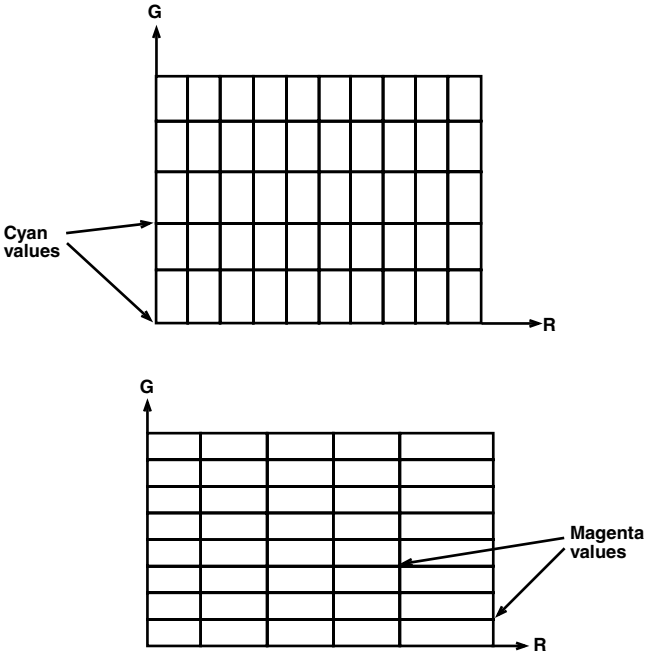


Figure 11.2 An anamorphic lattice in two dimensions.

way along a first dimension of a square and a fraction u of the way along a second dimension of the square, begin by computing two intermediate points, a fraction t along the edges that traverse the first dimension. Then compute the interpolant a fraction u of the way from the first to the second of those intermediate points. Either dimension may be interpolated first with identical results, as shown below.

t first:

$$p_0 = p_{00} + t(p_{01} - p_{00})$$
$$p_1 = p_{10} + t(p_{11} - p_{10})$$
$$p = p_0 + u(p_1 - p_0) = p_{00} + t(p_{01} - p_{00}) + u(p_{10} + t(p_{11} - p_{10}) - (p_{00} + t(p_{01} - p_{00})))$$
$$= p_{00} + t(p_{01} - p_{00}) + u(p_{10} - p_{00}) + tu(p_{11} - p_{10} - p_{01} + p_{00})$$

u first:

$$p'_0 = p_{00} + t(p_{10} - p_{00})$$
$$p'_1 = p_{01} + u(p_{11} - p_{01})$$
$$p = p'_0 + t(p'_1 - p'_0) = p_{00} + u(p_{10} - p_{00}) + t(p_{01} + u(p_{11} - p_{01}) - (p_{00} + u(p_{10} - p_{00})))$$
$$= p_{00} + t(p_{01} - p_{00}) + u(p_{10} - p_{00}) + tu(p_{11} - p_{10} - p_{01} + p_{00})$$

Figure 11.3 shows this operation for the case where the t interpolation is done first.

Triangular interpolation, applied to right triangles with two axis-aligned edges, employs the additional assumption that the function being

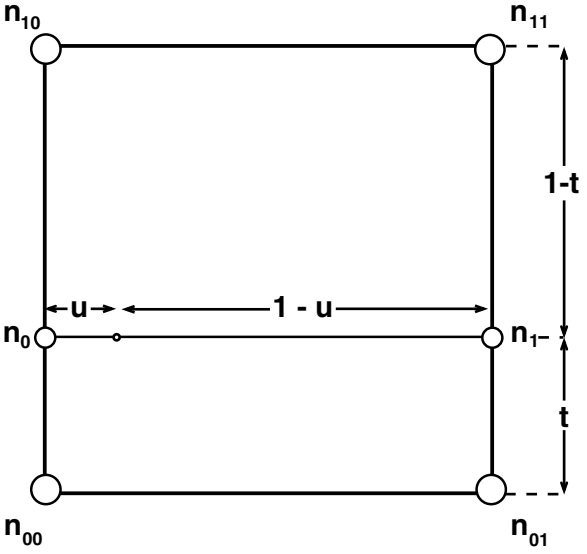


Figure 11.3 Bilinear interpolation.

approximated has zero cross derivatives. That is, the derivative with respect to u is the same irrespective of the value of t. With reference to Figure 11.4, given nodes n_{00}, n_{01} and n_{11} with values p_{00}, p_{01} and p_{11} corresponding to $t = u = 0$; $t = 0$, $u = 1$; and $t = u = 1$, respectively, begin with p_{00}, add $(p_{01} - p_{00})u$, to find a value corresponding to u of the way from p_{00} to p_{01}. Then add $(p_{11} - p_{01})t$ to find a value corresponding to t of the way from there to the line containing p_{11}. If both of the assumptions — linear function with zero cross derivatives — hold, the result of triangular interpolation is the same as that of bilinear.

$$
\begin{aligned}
p_0 &= p_{00} + t(p_{01} - p_{00}) \equiv p_{00} + t(p_{11} - p_{01}) \\
p_1 &= p_{10} + t(p_{11} - p_{01}) \\
p &= p_0 + u(p_1 - p_0) = p_{00} + t(p_{11} - p_{01}) + u(p_{10} + t(p_{11} - p_{01}) - (p_{00} + t(p_{11} - p_{01}))) \\
&= p_{00} + t(p_{11} - p_{01}) + u(p_{01} - p_{00})
\end{aligned}
$$

11.2.3 *Three-dimensional interpolation geometries*

Three-dimensional interpolation geometries include trilinear (all nodes), prism, pyramidal, and tetrahedral, in order of decreasing numbers of nodes used. The vertices of the interpolation solids for the various geometries are given in Table 11.1, with the labeling convention used in Figure 11.5. In all of these geometries, the facets of the solids within which interpolation is performed are either squares or triangles, hence the reliance on bilinear and triangular interpolation. From a computational standpoint, interpolating within a square is more expensive than interpolating within a triangle; this

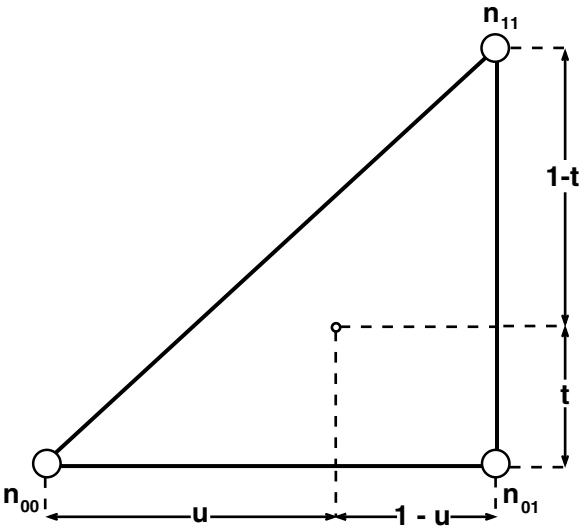


Figure 11.4 Triangular interpolation.

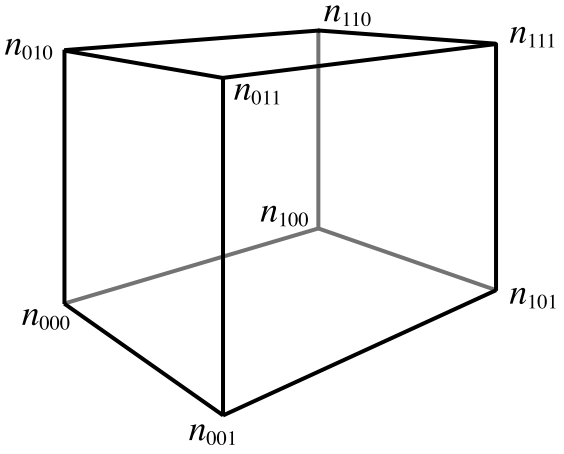


Figure 11.5 Rectangular volume showing vertex labels used in Table 11.1.

is one of the motivations of the various geometries. At the same time, the fewer cutting planes subdivide the cube, the fewer tests are required to select the interpolation solid. From a qualitative standpoint, geometries that use more nodes will perform stronger local averaging of the function being approximated, while geometries with fewer nodes will more closely follow the curvature of the function. The relative performance of the geometries will thus depend on the nature of the function and noise characteristics. The two most commonly used geometries are trilinear and tetrahedral — the

Table 11.1 Vertices Used for the Various Interpolation Geometries; Vertex Labels According to Figure 11.5

Trilinear	Prism	Pyramidal	Tetrahedral
All	n_{000}, n_{001}, n_{010}, n_{011}, n_{100}, n_{101}	n_{000}, n_{011}, n_{100}, n_{101}, n_{111}	n_{000}, n_{010}, n_{100}, n_{101}
	n_{010}, n_{011}, n_{100}, n_{101}, n_{110}, n_{111}	n_{000}, n_{001}, n_{010}, n_{011}, n_{101}	n_{000}, n_{001}, n_{010}, n_{101}
		n_{010}, n_{011}, n_{101}, n_{110}, n_{111}	n_{001}, n_{010}, n_{011}, n_{101}
			n_{010}, n_{100}, n_{101}, n_{110}
			n_{010}, n_{101}, n_{110}, n_{111}
			n_{010}, n_{011}, n_{101}, n_{111}

three-dimensional analogies of bilinear and triangular — representing the two extremes. Each of the geometries is described in detail below.

11.2.3.1 *Trilinear interpolation*

Trilinear interpolation uses all eight nodes and is the natural extension of linear interpolation to three dimensions. This is shown in Figure 11.6. To compute the interpolant at fractions t, u, and v along three respective dimensions, first compute four intermediate points (n_{00}, n_{01}, n_{10}, and n_{11}) that are t of the way along the edges varying in the first dimension. Then use those four points to define a square and perform bilinear interpolation as described in the previous section. From the square, two more points are computed (n_0 and n_1); these are used to compute the final point n.

Using the notation $L^n(x_1 \ldots x_n)$ to refer to linear interpolation in n dimensions, $L^3(t, u, v)$ then denotes trilinear interpolation. It may be expressed recursively in terms of lower-order interpolation as:

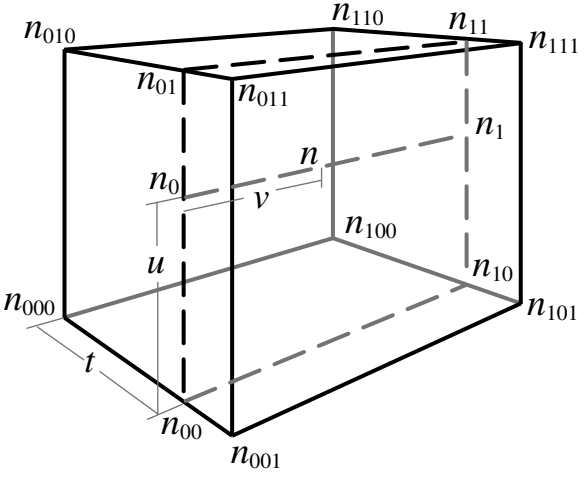


Figure 11.6 Trilinear interpolation.

$$
\begin{aligned}
L^3(t, u, v) &= L^2{}_0\,(t, u) + (L^2{}_1(t, u) - L^2{}_0\,(t, u))\; v \\
&= L^1{}_{00}\,(t) + (L^1{}_{01}(t) - L^1{}_{00}(t))u + \{L^1{}_{10}\,(t) + (L^1{}_{11}\,(t) - L^1{}_{10}(t))\; u \\
&\quad - [L^1{}_{00}\,(t) + (L^1{}_{01}\,(t) - L^1{}_{00}(t))u]\}\; v \\
&= p_{000} + (p_{001} - p_{000})\; t + (p_{010} + (p_{011} - p_{010})\; t - (p_{000} + \\
&\quad (p_{001} - p_{000})\; t))\; u + \{p_{100} + (p_{101} - p_{100})\; t + (p_{110} + \\
&\quad (p_{111} - p_{110})\; t - (p_{100} + (p_{101} - p_{100})\; t))\; u - [p_{000} + (p_{001} - p_{000})\; t \\
&\quad + (p_{010} + (p_{011} - p_{010})\; t - (p_{000} + (p_{001} - p_{000})\; t))\; u]\}\; v
\end{aligned}
\tag{11.3}
$$

Computationally,

$$
L^3(t, u, v) = p_{000} + \Delta^t{}_{00}\; t + \delta^{uv}{}_0\; u + (\Delta^v + \Delta^{tv}{}_0\; t + \delta^{uv}{}_1\; u)\; v
$$

$$
\begin{aligned}
&\Delta^t{}_{00} = p_{001} - p_{000}; && \Delta^t{}_{01} = p_{011} - p_{010}; && \Delta^t{}_{10} = p_{101} - p_{100}; && \Delta^t{}_{11} = p_{111} - p_{110}; \\
&\Delta^u{}_0 = p_{010} - p_{000}; && \Delta^u{}_1 = p_{110} - p_{100}; && \Delta^v = p_{100} - p_{000}; && \Delta^{tu}{}_0 = \Delta^t{}_{01} - \Delta^t{}_{00}; \\
&\Delta^{tu}{}_1 = \Delta^t{}_{11} - \Delta^t{}_{10}. && \Delta^{tv}{}_0 = \Delta^t{}_{10} - \Delta^t{}_{00}. && \Delta^{uv}{}_0 = \Delta^u{}_1 - \Delta^u{}_0. && \Delta^{tuv} = \Delta^{tu}{}_1 - \Delta^{tu}{}_0, \\
&\delta^{uv}{}_0 = \Delta^u{}_0 + \Delta^{tu}{}_0\; t. && \delta^{uv}{}_1 = \Delta^{uv}{}_0 + \Delta^{tuv} t
\end{aligned}
\tag{11.4}
$$

This requires seven multiplications and seven additions, with twelve precomputed coefficients stored at each node.

11.2.3.2 *Prism interpolation*

Prism interpolation splits the cube in half along a face-diagonal, as shown in Figure 11.7. When interpolating in RGB or like color spaces, the cutting plane should be chosen to contain the neutral (R = G = B) axis so that

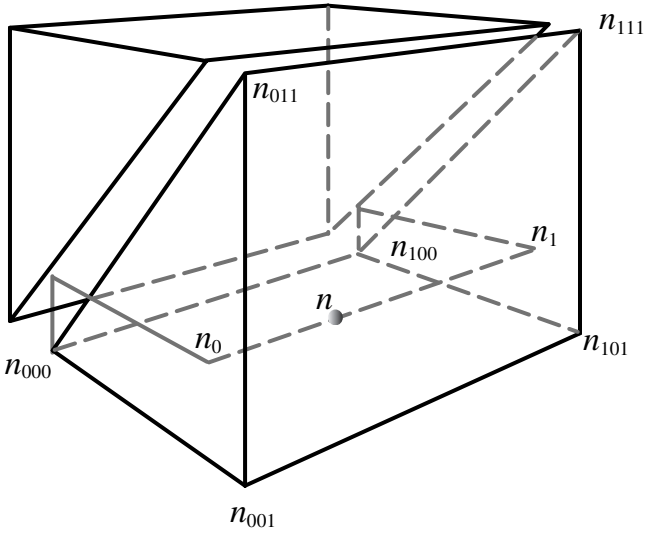


Figure 11.7 Prism interpolation.

interpolation along this axis can be carefully controlled. An initial test is required to determine in which prism the point lies, and then interpolation is performed using only six, rather than all eight, points of the cube. As shown in Figure 11.7, the point is first projected onto each of the two triangular faces, and triangular interpolation is performed within these faces using fractional distances t and u. The resulting two points are interpolated linearly using the fractional distance v. These two steps can be combined as

$$P^3{}_{t<u}(t, u, v) = p_{000} + (p_{011} - p_{001})\, u + (p_{001} - p_{000})\, t + [p_{100} + (p_{111} - p_{101})\, u + (p_{101} - p_{100})\, t - (p_{000} + (p_{011} - p_{001})\, u + (p_{001} - p_{000})\, t)]\, v \tag{11.5}$$

Computationally,

$$\mathrm{P}^3{}_{t<u}(\mathrm{t, u, v}) = s_5 + [p_{100} + s_2 + s_4 - s_5]\, v \tag{11.6}$$

where

$$s_1 = (p_{011} - p_{001})\, u = \Delta_1 u \quad s_2 = (p_{111} - p_{101})\, u = \Delta_2 u \quad s_3 = (p_{001} - p_{000})\, t = \Delta_3 t$$
$$s_4 = (p_{101} - p_{100})\, t = \Delta_4 t \quad s_5 = p_{000} + s_1 + s_3$$

Prism interpolation requires five multiplications and six additions with nine coefficients stored at each node. The reduced multiplications come at the expense of a single test.

11.2.3.3 Pyramidal interpolation

Figure 11.8 shows the geometry of pyramid interpolation. It uses one corner (which for RGB-like spaces should be either the node [X][Y][Z] or [X+1][Y+1][Z+1]) as the apex of three pyramids, the bases being the three faces that share the diagonally opposite vertex. Two tests are required to determine in which pyramid the point lies, and then interpolation is performed using five points of the cube.

$$P^3{}_{t<u,t<v}(t, u, v) = p_{000} + s_1\, u + s_2\, t + (s_3 + s_4\, u)\, v \tag{11.7}$$

where $s_1 = p_{010} - p_{000}$, $s_2 = p_{111} - p_{110}$, $s_3 = p_{100} - p_{000}$, $s_4 = p_{110} - p_{100} - p_{010} + p_{000}$

$$P^3{}_{u>v,t\geq v}(t, u, v) = p_{000} + s_1\, v + s_2\, u + (s_3 + s_4\, v)\, t \tag{11.8}$$

where $s_1 = p_{100} - p_{000}$, $s_2 = p_{111} - p_{101}$, $s_3 = p_{001} - p_{000}$, $s_4 = p_{101} - p_{001} - p_{100} + p_{000}$

$$P^3{}_{u>v,t\geq v}(t, u, v) = p_{000} + s_1\, t + s_2\, v + (s_3 + s_4\, t)\, u \tag{11.9}$$

where $s_1 = p_{001} - p_{000}$, $s_2 = p_{111} - p_{011}$, $s_3 = p_{010} - p_{000}$, $s_4 = p_{011} - p_{010} - p_{001} + p_{000}$

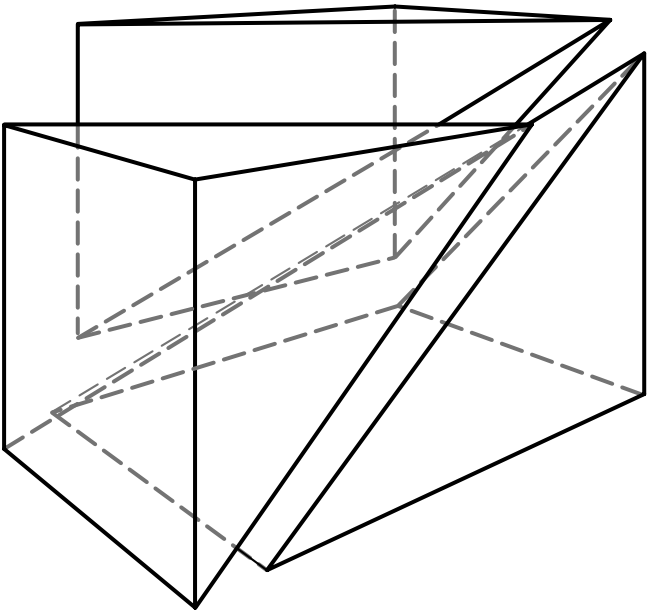


Figure 11.8 Pyramidal interpolation.

Pyramidal interpolation requires four multiplications, with five values stored at each node.

11.2.3.4 Tetrahedral interpolation

Due to its computational simplicity, tetrahedral interpolation is a very popular choice for three-dimensional color transformations.[2] The most common form divides the two prisms of prism interpolation into three tetrahedra each, as shown in Figure 11.9. All six tetrahedra share one common edge: the main diagonal of the cube. In the case of RGB spaces, the diagonal axis should coincide with the neutral (R = G = B) axis. The remaining edges of the tetrahedra join the neutral vertices to vertices diagonally opposite them on shared faces. Tetrahedral interpolation requires, on average, 2.5 comparison tests (i.e., 2 tests or 3 tests, both with 50% probability) to find the enclosing tetrahedron. The code structure is given below:

```
If t < u // lower front prism
    If t > v
        Interpolate within tetrahedron 1
    Else
        If u < v
            Interpolate within tetrahedron 2
        Else
            Interpolate within tetrahedron 3
```

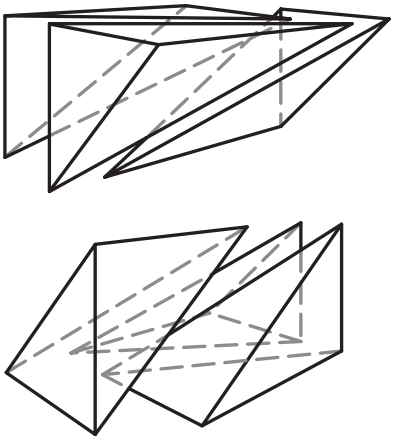


Figure 11.9 Tetrahedral interpolation.

```
Else // upper, back prism
   If t < v
      Interpolate within tetrahedron 4
   Else
      If u < v
         Interpolate within tetrahedron 5
      Else
         Interpolate within tetrahedron 6.
```

The case where t < u, t > v (i.e., tetrahedron 1) is shown in Figure 11.10. Interpolation among the four points within the enclosing tetrahedron is given by

$$P^3(t, u, v) = p_{000} + s_t\, t + s_u\, u + s_v v \tag{11.10}$$

The interpolation weights s_t, s_u, s_v depend on which tetrahedron contains the input point, and they are determined by comparing the magnitudes of t, u, and v, as shown above and in Table 11.2. Table 11.3 gives the differences Δ_i used in Table 11.2 (and hence the values of s_t, s_u, and s_v). The 12 differences correspond to the 12 edges of the cube. Assuming the differences are all precomputed, tetrahedral interpolation requires 3 additions and 3 multiplications, with 13 values stored at each node. It is essentially three triangular interpolations: one per dimension. On most processors, memory fetch times dominate in tetrahedral interpolation.

Other variants of tetrahedral interpolation exist; for example, it is possible to partition the rectangular volume into only five tetrahedra. Details are given Kang's book.[3]

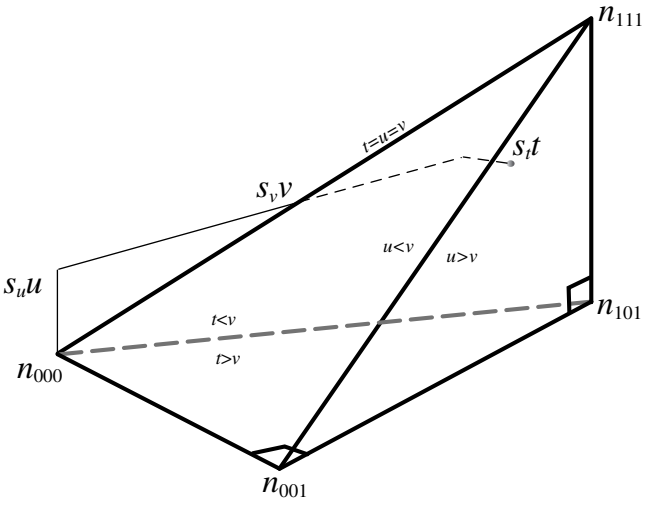


Figure 11.10 Sequential linear interpolation in two dimensions.

Table 11.2 Tetrahedral Interpolation: The Various Tests and Corresponding Interpolation Weights

Case			s_t	s_u	s_v
$t < u$	$t > v$		Δ_1	Δ_5	Δ_2
$t < u$	$t \leq v$	$u < v$	Δ_9	Δ_8	Δ_4
$t < u$	$t \leq v$	$u \geq v$	Δ_9	Δ_5	Δ_{11}
$t \geq u$	$t < v$		Δ_3	Δ_7	Δ_4
$t \geq u$	$t \geq v$	$u < v$	Δ_{10}	Δ_6	Δ_2
$t \geq u$	$t \geq v$	$u \geq v$	Δ_{10}	Δ_7	Δ_{12}

Table 11.3 Difference Quantities Used in Tetrahedral Interpolation

$\Delta_1 = p_{110} - p_{010}$,	$\Delta_2 = p_{111} - p_{110}$,	$\Delta_3 = p_{101} - p_{001}$,	$\Delta_4 = p_{001} - p_{000}$;
$\Delta_5 = p_{010} - p_{000}$,	$\Delta_6 = p_{110} - p_{100}$,	$\Delta_7 = p_{111} - p_{101}$,	$\Delta_8 = p_{011} - p_{001}$;
$\Delta_9 = p_{111} - p_{011}$,	$\Delta_{10} = p_{100} - p_{000}$,	$\Delta_{11} = p_{011} - p_{010}$,	$\Delta_{12} = p_{101} - p_{100}$.

11.3 *Interpolation on irregular lattices*

In the most general case where the lattice comprises an arbitrary set of points in three-dimensional space, it is possible to construct a tetrahedral partition of the space with these points using the DeLaunay tetrahedralization algorithm.[4] This generalizes for arbitrary dimensions as well. The advantage of this approach is the flexibility with which the lattice nodes can be positioned

in color space. However, the step of retrieving the enclosing tetrahedron becomes far more complex than in the case of regular lattices. Furthermore, the interpolation step is also more complex, involving tetrahedral inversion (see Chapter 5, Section 5.4.5). For this reason, the DeLaunay partition is rarely used in applications that require efficient real-time mapping through LUTs, and it will not be discussed further.

A special case of an irregular lattice is the sequential lattice, a structure that offers more flexibility than a regular lattice but at a modest increase in computational cost. The general sequential interpolation architecture is described in detail in Chapter 5 and will not be presented again here. We instead focus on the particular example of sequential linear interpolation (SLI).[5] The SLI lattice is shown in two dimensions in Figure 11.11. Along the first dimension c_1 are node levels $c_{1,1}$,…, $c_{1,N}$. At each such node level, a set of nodes is positioned along c_2. The interesting feature of SLI is that the node separation along c_2 can vary depending on the location along c_1. With reference to the diagram on the right in Figure 11.11, interpolation for an arbitrary input point P is performed as follows:

- Project P onto the c_1 axis and locate the two neighboring levels $c_{1,i}$, $c_{1,i+1}$.
- Project P onto the line $c_{1,i}$ and perform linear interpolation between points D_1 and D_2, resulting in intermediate value D_a.
- Project P onto $c_{1,i+1}$ and perform linear interpolation between D_3 and D_4, resulting in D_a.
- Perform 1-D interpolation between D_a and D_b to obtain the final output.

Note that this process is very similar to bilinear interpolation, described in Section 11.2.2. Indeed, if the node placements along c_2 are independent of c_1, the SLI lattice reduces to the standard regular lattice, and the algorithm reduces to bilinear interpolation. The more general and interesting case

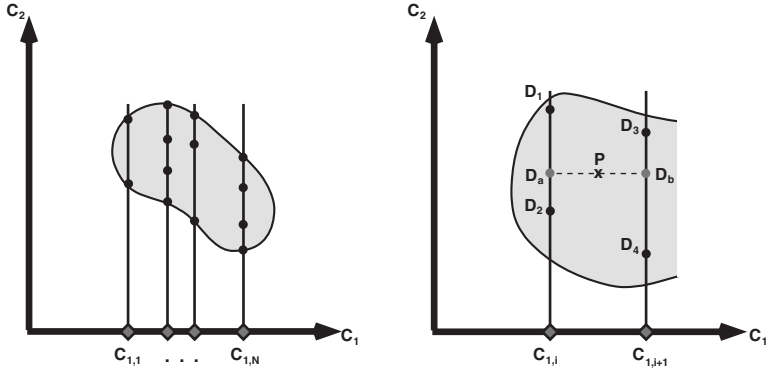


Figure 11.11 Sequential linear interpolation in two dimensions.

where interpolation along c_2 depends on both c_2 and c_1 enables a more efficient placement of nodes in those regions where the function being approximated exhibits greatest curvature. Optimal design of an SLI lattice to suit the characteristics of the underlying transformation is described in detail by Chang et al.[6] and Agar et al.[7]

The algorithm just described extends readily to an arbitrary number of dimensions, with node level placement along the ith dimension depending on the input values along the previous $i - 1$ dimensions. In the common case of three dimensions, SLI requires seven multiplications and seven additions and is hence computationally equivalent to trilinear interpolation. However, SLI requires prestoring more fractional quantities. This is because two of the quantities t, u, v in Equations 11.3 and 11.4 vary depending on the position of the input point in the 3-D table. Normally, these are precomputed as described in Section 11.4. The chief advantage of SLI lies in its more efficient use of table space, allowing a table of a given number of nodes to more accurately follow the curvature of the function being approximated.

11.4 Acceleration techniques

Lookup table interpolation is much faster than inverting measured data or evaluating a printer model. However, applying the techniques below may speed it up further. Not all acceleration techniques are applicable to all applications; some are architecture dependent, while the advantage of others depends on the nature of the input. Each should be evaluated in the context of the intended application.

11.4.1 Caching node values

For some machines, three-dimensional lookup is expensive, and if 13 values are stored at a node, pulling in these values from memory only adds to the expense. In such cases, a small cache of node values may be beneficial. Typical images contain long runs (tens or even hundreds of pixels) of colors similar enough to be within a node of each other. A cache containing seven nodes is generally sufficient for such purposes. In such an implementation, the cache is checked before the full three-dimensional lookup is performed. On most modern architectures, these values will be cached in the hardware cache. Artificially checking a cache may result in reduced performance in the presence of a hardware cache. One might consider reducing the precision of the values stored at the nodes if the hardware cache is too small to hold seven nodes.

11.4.2 Caching output values

For most machines, the computations, even for tetrahedral interpolation, are more expensive than simple caching schemes. The challenge lies in making the amortized cost of cached interpolation less expensive than reinterpola-

tion. Some number n_c of input–output pairs is stored in a table, with a simple hashing scheme determining the location in the table at which the color pair containing input color c_i is stored.

To convert a new color, apply the hash function $h(n_c, c_i)$ to obtain an address in the table. If the color pair containing c_i is in that location, return the output color of that pair. Otherwise, perform the normal interpolation function, and store the input and output colors in a color pair at the computed location.

The value of caching output values depends on colors repeating before they are overwritten in the table. Only large/high-resolution images are important to the discussion of efficiency. These images typically have many regions at least 0.1% of their size containing nearly constant colors. Even highly textured regions typically vary only along a single line in color space. In a 300-dpi scan of a 4×5 inch image, a region of 0.1% of the image is 1800 pixels. Due to film grain, scanner noise, and other sources of noise, even a region of apparently constant color will not have exactly constant colors throughout such a region. However, it is not uncommon to find regions of this size containing fewer than 300 distinct colors.

The size of the table used for caching should typically be at least large enough to hold a scanline or two of the typical image. The chance of reusing a color increases substantially if the reuse may happen anywhere in a multiscanline region rather than in a single scanline. Very large tables have two disadvantages: besides requiring a large quantity of memory, successively needed values are unlikely to be in the hardware cache because, by design, similar values are scattered throughout the table.

11.4.3 Hashing

Hashing is discussed at length in Aho, Sethi, and Ullman.[8] Conventionally, collisions (when a second value maps to an already-used location) are handled using linked lists or secondary hash schemes. For caching color conversions, it is more efficient simply to replace the old value with the newly computed one, as described above. An ideal hash function will map all the distinct colors found in a few scanlines of an image to different locations in the table. Designing such a perfect function requires knowledge of the colors in the image. However, a few guiding principles can be used to design good enough hash functions.

- Use only additions, shifts, and logical operations.
- Assume that the few high-order bits of each separation will be constant (hence, they contribute no new information).
- Assume that variation is primarily in luminance (RGB variations are correlated; L*a*b* variations decorrelated).

A simple hash function based on a mapping *hashpjw* given in Aho et al.[8] for 8-bit RGB color is $h(n_c, c_i) = ((((r_i << 4) + g_i) << 4) + b_i) \ \% \ n_c$.

11.4.4 Precomputing fixed-point quantities

To compute any of Equations 11.2 through 11.10 requires computing values of t, u, and v, normally in floating point, and multiplying them by the appropriate differences. For example,

$$X = \lfloor x\sigma \rfloor, \quad t = x\sigma - X \tag{11.11}$$

The value of σ is a constant for a given lookup table size; a table of C_{max} entries will give the values of X and t in Equation 11.11. This sacrifices no precision, as the subsequent operations may still use floating point.

The basic operation of linear interpolation involves multiplying a difference by a parameter in the [0, 1) range and then adding it to a base value. Essentially the same operations appear throughout Equations 11.2 through 11.10.

$$\text{Lerp}(a, b, t) = a + (b - a)\, t \tag{11.12}$$

Values a and b may be viewed as being fixed-point numbers with n bits fraction. In this representation, they range from 0 to $1 - 1/C_{max}$. Converting Equation 11.12 from floating point to fixed point with Z fraction bits,

$$\text{Lerp}(a, B, T) = a + ((D\ T + (1 << (Z - 1))) >> Z) \tag{11.13}$$

with $D = (b - a)$, $T = \lfloor t(1<<Z) + 0.5 \rfloor$.

All of the variables in Equation 11.12 are originally retrieved from lookup tables. Hence, replacing the lookup table entries with the corresponding values in Equation 11.13 eliminates all of the floating to fixed point conversions from the interpolation. On typical machines, conversions to and from floating point are expensive, so this is a significant savings. All of the fixed point values in Equation 11.12 have known ranges: $0 \le a < 1$, $-1 + 1/C_{max} \le b - a \le 1 - 1/C_{max}$, $0 \le t < 1$. With $C_{max} = 2^8$, a value of $Z = 23$ will not cause overflow on a 32-bit machine in the expression $(D\ T) >> Z$.

11.4.5 Eliminating multiplications

It is possible to eliminate all of the multiplications at the expense of an extra table lookup. Whether this speeds up the interpolation will depend on the machine architecture. On a typical RISC machine, multiplications are implemented as multiple instruction microcode subroutines, making them more expensive than table lookups. On a machine with relatively slow memory and a fast multiply instruction, leaving in the multiplications may be faster.

In Equation 11.13, D takes one of $2\,C_{max} - 1$ possible values. It is multiplied by T, which takes on one of C_{max} values. T itself and D are retrieved from lookup tables; there is no reason why the value in the table from which T was retrieved should not, instead, contain a pointer to a multiplication table,

or an index into a table of multiplication tables. Assuming $n = 8$ bits, there would be 256 tables of 511 entries, for a total of just under 128k entries. The entries themselves could be either short words (to save space) or full words (for faster computation).

The expression then becomes

$$\text{Lerp}(a, D, T) = a + ((\text{M}[D, T] + (1 << (\text{Z} - 1))) >> Z) \tag{11.14}$$

with $M[j, k] = j\ k$, and with the precomputed values of D offset by C_{max} so that the table indices are always positive. The rounding and final shift can also be folded into the table; however, for tetrahedral interpolation, there are three differences multiplied by three parameters and added to a node value. In that case, $(1 << (Z - 1)$, should be added only once, as in

$$\text{Tetra}(a, B, C, D, T, U, V) = a + ((\text{M}[B, T] + \text{M}[C, U] + \text{M}'[D, V]) >> Z) \tag{11.15}$$

with $\text{M}'[j, k] = \text{M}[\text{j}, k] + (1 << (\text{Z} - 1))$.

11.4.6 Eliminating tests

On modern machines, frequent branches are expensive because they require the instruction pipeline to be flushed. Some machines have a large enough instruction cache so that instructions may be fetched for both sides of a branch, up to two or even three branches, in which case the penalty is negligible until the number of branches without intervening computation reaches three or four. Thus, it is sometimes, but not always, advisable to eliminate tests. For tetrahedral interpolation, recall from Section 11.2.3.4 that tests are required to determine one of six enclosing tetrahedra.

In all six cases, the expression to be evaluated is the same; only the operands s_t, s_u, and s_v differ. These are taken from a set of 12, depending on the case. If a node is defined as containing the value of p_{00} and an array containing the 12 edge-differences, the case analysis selects which of the 12 array indices are used to obtain s_t, s_u, and s_v. Each of the tests in the case analysis is decided on the basis of the sign of the difference between two of t, u, and v. Therefore, the case analysis may be replaced by

$$\text{Dtu} = \text{t} - \text{u};\ \text{Dtv} = \text{t} - \text{v};\ \text{Duv} = \text{u} - \text{v} \tag{11.16}$$

$$\text{Selector} = ((\text{Dtu} >> \text{s}) \,\&\, 1) \,|\, ((\text{Dtv} >> (\text{s} - 1)) \,\&\, 2) \,|\, ((\text{Duv} >> (\text{s} - 2)) \,\&\, 4)$$

where s is the number of bits in the machine word holding Dtu, Dtv, and Duv. An eight-entry lookup table of three-tuples will give the indices of s_t, s_u, and s_v. This completely replaces the tests with straight-line code, making the code smaller and possibly faster. Whether the code is faster depends on whether the extra shifts and *or*s of values likely to be in registers is more or

less expensive than the average 2.5 tests. The shift and mask combination can also be incorporated into a LUT, which may help as well. This depends on the trade-off between two ALU operations and a memory fetch. Fully optimized, the version with 2.5 tests ran in approximately 71 ns per color on a 500-MHz PowerPC G3; the best version without tests took about 79 ns, 10% slower.

11.4.7 Data Formats

The format of the data in the table may make as much difference as many of the other acceleration techniques. Most of the processing involved in transforming large images is memory-bound. That is, the time required on current architectures tends to be dominated by the number of memory accesses. Typical processors have 32- or 64-bit memory buses, which means that a 32-bit (or 64-bit) memory fetch costs no more than a single-byte fetch. With this in mind, an image stored as packed pixels will generally be more efficient to process than one with separately stored channels. Similarly, a multidimensional LUT containing all of the data for a color at each node enables more efficient processing than one table per output separation.

Many processors penalize conversion to and from floating point. Lookup tables are typically computed in floating point, but color images are most commonly stored as arrays of fixed-point integers. Output devices invariably require integer-valued pixels. Hence, the LUT should be converted to fixed point once and then all pixel operations carried out in fixed point.

Besides memory access time, packed pixels provide acceleration opportunities, as multiple additions or averaging operations may be carried out in parallel across separations. Specialized media processing instructions (VIS instructions on the SPARC processor, MMX on Intel, AltiVec on PowerPC) often permit the operations for a linear interpolate (LERP, above) to be computed in parallel on two or four sets of operands (with the same value of the parameter) in the time normally required to do one such operation, provided the operands are formatted appropriately.

11.5 Color transforming palettized images

A palettized image is one in which pixel colors are limited to a relatively small palette of colors. A common palette size is 256, used in 8-bit displays. The process of designing the optimal palette and assigning image pixels to the most appropriate palette color is commonly referred to as *color quantization* and is covered in detail in another chapter. The most efficient way to apply a color transformation to a palettized image is to apply the transformation *a priori* to the palette colors and store the palette in the new transformed color coordinates. The color transformation problem is thus translated to one of color quantization.

11.6 Subsampled color correction

11.6.1 Introduction

Some applications use luminance/chrominance-aligned color spaces. For example, YCrCb allows color/brightness/saturation adjustments at the user interface, and the color fax standard uses L*a*b*, which is also luminance/chrominance based. To avoid ambiguity with yellow, assume that the input is a luminance/chrominance space, denoted LCrCb, a surrogate for other luminance/chrominance spaces.

It is well known that the visual system's spatial resolution for chrominance information is less than its resolution for luminance information. This fact is exploited both in TV transmission (where Cr and Cb are transmitted with half the bandwidth of L) and in JPEG/ADCT compression. Good quality color conversion may be had, therefore, by selecting one pixel from every 2×2 block for full conversion through the 3-D LUT, and attempting to approximately preserve only the luminance channel for the remaining pixels.

In the case where the output space is RGB (or CMY), this consists of the following steps:

- From the input block {(L00, Cr00, Cb00), (L01, Cr01, Cb01), (L10, Cr10, Cb10), (L11, Cr11, Cb11)}, select one pixel, (L_m, Cr_m, Cb_m) as the master pixel.
- Convert this pixel using full 3-D LUT interpolation to (R_m, G_m, B_m).
- For each pixel P_{ij} in the block, assign

$$R_{ij} = (R_m/L_m)L_{ij};\ G_{ij} = (G_m/L_m)L_{ij};\ B_{ij} = (B_m/L_m)L_{ij.} \tag{11.17}$$

For efficiency, the loop need only be iterated over the remaining three pixels; however, there may be advantages in hardware to performing it in parallel on all four.

Note that the ratios in Equation 11.17 are invariant over pixel and need only be computed once per block. Alternatively, the ratio L_{ij}/L_m is invariant over separation and need only be computed once per pixel. The multiplications and divisions in Equation 11.17 can be sped up by computing in log space.

$$R_{ij} = a^{\log_a R_m - \log_a L_m + \log_a L_{ij}} \tag{11.18}$$

The difference $\log_a R_m - \log_a L_m$ is invariant within the block and should be computed only once. Both the exponentiation and logarithms can be performed by one-dimensional table lookup. The loss of precision that results from computing in eight-bit log space should not be a problem because the

space is density-like and mimics the logarithmic behavior of the visual system.

The simplest and probably most reliable choice of the master pixel is the one with maximum luminance. Then, all of the ratios in Equation 11.17 are between 0 and 1. One can conceive of an analogous rule for converting to CMYK. However, four-colorant rendering typically involves some type of undercolor removal (UCR) and gray component replacement (GCR) strategy, establishing the relative proportions of CMY and K used to produce a given color. UCR/GCR can be very different from printer to printer; hence, a simple application of the above method may not always work. For the case where there is very little CMY subtraction, the following may be used:

- Convert one pixel from (L_m, Cr_m, Cb_m) to (C_m, M_m, Y_m, K_m) using the full three-dimensional lookup table approach.
- For the remaining pixels, P_{ij}, set

$$C_{ij} = 1 - ((1 - C_m)/L_m)L_{ij} \tag{11.19}$$

  with analogous expressions for M_{ij} and Y_{ij}.
- The same rule should not be used for K_{ij} because this would result in a situation where K is introduced at an edge between two colors, neither of which contains K. Instead one might simply assign $K_{ij} = K_m$.

Because L_m is the maximum L, the ratio is always in the 0, 1 range, so all of the colors are between 0 and their original values.

At the expense of six more additions per block (and two shifts in software), Cr_m and Cb_m for the master pixel can be computed as the mean values of Cr_{ij} and Cb_{ij}, providing substantial improvements in color image quality and reducing the tendency of the method to erode color against white edges. In the JPEG architecture, the chrominance channels are subsampled after filtering; hence, Cr and Cb are already averaged, and this step is not needed.

11.6.2 Results

Subsampled color correction was compared to conventional color correction, both in terms of computational cost and image quality. For the simulation, subsampled color conversion was performed from LCrCb to CMY using the six previously described steps, with no chroma averaging. This was followed by UCR/GCR and tone reproduction curve (TRC) correction applied to the full resolution image. The standard used was conventional color conversion from LCrCb to CMYK using three-dimensional lookup and tetrahedral interpolation. Table 11.4 summarizes the computational requirements per pixel. Clearly, subsampled color correction yields substantial savings in multiplications, additions, and shifts, with moderate savings in lookups — and an extra comparison operation. When full resolution and subsampled color correction are compared on 400-dpi prints, the two images are visually almost indistinguishable.

Table 11.4 Computational Savings from Subsampled Color Correction

Method	Multiples	Adds	Compares	1-D table lookups	Binary shifts
Conventional	12	14	2.5	19	2
Subsampled	2.25	9.75	3.63	17.1	0.5
Percent savings	81	30	–45	11	75

11.7 Color transforming JPEG compressed images

Many high-end printing systems store and/or transmit compressed images to save bandwidth, memory, and disk space. Prior to printing, the system will decompress and then color correct the image to compensate for the characteristics of a particular printer. In the conventional architecture, the decompression step converts the image from some encoded format (e.g., JPEG with Huffman encoding) into the spatial domain. In this domain, the color image may be stored in some standard device-independent color space such as IJG/CCIR YCrCb, SMPTE RGB, or Kodak Photo Ycc. The ensuing color correction transforms colors from this space to device coordinates. It is more efficient, however, to perform the color space conversion in the compressed domain.[9] As with subsampled color correction, this approach involves parsing color correction into two processes, applying an expensive process to a low-resolution version of the image before decompression and an inexpensive operation to the entire image after decompression. The ratio of resolutions controls the trade-off between quality and speed.

In this section, we focus on color transforming JPEG compressed images. JPEG compression is described in detail in Chapter 8 and therefore is not presented here.

11.7.1 Correcting for RGB devices

RGB devices include monitors (cathode-ray tube based and liquid crystal display based), film recorders, and some printers that present an RGB interface. To perform fast color correction on compressed images, separate the color correction problem into two phases as shown in Figure 11.12a. First, account for any mismatch between the image and device primaries, and interactions among the device primaries. Second, linearize the individual primaries to some measured standard. (In Chapter 5, these two phases are denoted *characterization* and *calibration*, respectively.) The first phase converts device-independent data to device-dependent linearized data, and it requires at least a 3×3 matrix multiplication, and possibly a full three-dimensional table lookup with interpolation. The second phase involves using tone reproduction curves (TRCs) to linearize the device. The TRCs are implemented as one-dimensional table lookups and are therefore cheaper than the first phase. Normally, if the first phase involves a three-dimensional table lookup

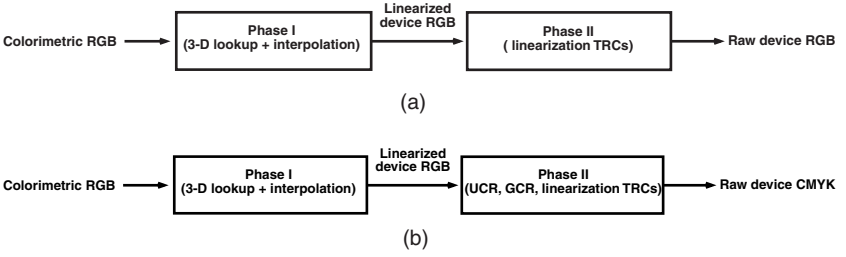


Figure 11.12 The two phases of color correction for (a) RGB and (b) CMYK devices.

with interpolation, the second phase can be combined into the same table lookup, yielding the entire result. When converting compressed images, it is advantageous to keep the two phases separate.

For reasons that will become apparent shortly, it is important that the first phase preserves the sense of the color space. That is, if the input image data is in a device-independent RGB space (e.g., sRGB), this will be converted to a device-dependent RGB space. Similarly, if the input is in a device independent luminance-chrominance form (e.g., YCbCr), it will be converted to a device-dependent luminance chrominance space, $Y'C_1'C_2'$. Note that the relation between YCbCr and RGB need not be kept between $Y'C_1'C_2'$ and the device-dependent RGB space (i.e., $R'G'B'$). The advantage of choosing a suitable luminance-chrominance space $Y'C_1'C_2'$ is to reduce operations when converting data to $R'G'B'$, as compared to the standard conversion from YCbCr to RGB. The disadvantage is the possibility of having data in different color spaces for the same DCT block (to be discussed later). The device luminance-chrominance space is defined by a simple transformation from device RGB (or CMY),

$$\begin{aligned} Y' &= 3\,R'/8 + G'/2 + B'/8 \\ C_1' &= B' - Y' \\ C_2' &= R' - Y' \end{aligned} \tag{11.20}$$

The inverse transform from device $Y'C_1'C_2'$ to device RGB is simply

$$\begin{aligned} R' &= C_2' + Y \\ G' &= Y - 3\,C_2'/4 - C_1'/4 \\ &= Y - C_2' + (C_2' - C_1')/4 \\ B' &= C_1' + Y \end{aligned} \tag{11.21}$$

which involves a total of five additions and one shift. Linearization TRCs are then applied to the device RGB values prior to output.

11.7.2 *Correcting for CMYK Devices*

For four-colorant printers, color correction comprises a transformation from a device-independent three-tuple to printer CMYK. Again, the transformation can be divided into two phases. The first phase converts the device-independent three-tuple (e.g., colorimetric RGB) into a device-dependent three-tuple with the same color coordinate orientation (e.g., printer RGB). This transformation is invariably implemented by a three-dimensional LUT and is the more computationally intensive phase. As before, the second phase comprises one-dimensional linearization TRCs. Additionally, four-colorant printers require undercolor removal (UCR) and gray component replacement (GCR), converting printer RGB to printer CMYK. (See Chapter 5 for details on UCR and GCR.) Figure 11.12b summarizes the two phases. Often, both phases are combined into a single LUT that maps the device-independent three-tuple directly to CMYK.

11.7.3 *Color correction in the JPEG compressed domain*

In conventional applications, the image is first decompressed, and then the color correction transform is applied, as shown in Figure 11.13. It is often faster to combine the decompression and color correction. We will describe the CMYK case; the RGB is simpler and easy to derive from the description of the CMYK case. We assume that images are stored in colorimetric (device-independent) RGB space and JPEG compressed format. We exploit the fact that the human visual system is most sensitive to errors at low spatial frequencies. (This is especially true of chromatic errors.) We thus propose an efficient color correction scheme that performs the more expensive (Phase I) correction on only a sub-image corresponding to the first few low-order DCT coefficients. The simpler (Phase II) color correction is then performed on the entire image. The steps are summarized below:

- Obtain the $N \times N$ sub-block of low-order DCT components, $N < 8$.
- Perform the $N \times N$ inverse DCT, and obtain a reduced block of an RGB image of reduced resolution.
- Apply Phase I color correction to the pixels of the $N \times N$ reduced block by three-dimensional lookup and interpolation, yielding an image in printer RGB coordinates.
- Perform a forward $N \times N$ DCT on the reduced block, and place the resulting DCT components into their respective positions in the original 8×8 DCT block.

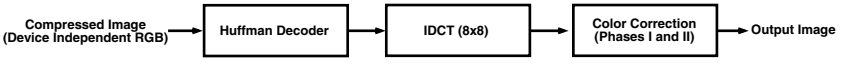


Figure 11.13 Standard approach of color correcting JPEG compressed images.

- Perform an 8 × 8 inverse DCT to the resulting block to reconstruct the image at full resolution.
- Apply Phase II color correction to all image pixels of the resulting image.

These steps are summarized in Figure 11.14. Note that because the N × N sub-block of partially corrected pixels is reinserted into the 8 × 8 block in the fourth step, it is important that Phase I correction preserve the orientation of color space (i.e., colorimetric RGB to device RGB, colorimetric YC_1C_2 to device YC_1C_2, etc.).

In the luminance/chrominance case, we use colorimetric YCbCr and device YC_1C_2. As YC_1C_2 is just a simplification of YCbCr, we expect the two color orientations to be very close. The DCT block right before inverse DCT contains low-frequency components from device YC_1C_2 but high-frequency components from colorimetric YCbCr. As the difference between colorimetric and device YC_1C_2 (or YCbCr) is generally more significant than the differences between colorimetric YC_1C_2 and colorimetric YCbCr, the slightly different high-frequency components (whenever they are present) do not affect the quality of the color reproduction. Because the lowest frequency term (DC) is always converted using the full process, every block has the correct color *on average.*

11.7.4 Results

Results indicate significant computational savings over the conventional approach (i.e., full decompression, followed by RGB to CMYK by three-dimensional LUT and interpolation for all image pixels), with little loss in image quality. The quality/cost trade-off clearly depends on the number of low-order DCT coefficients (i.e., N) to which Phase I color correction is applied. Experiments were carried out for N = 1, 2, 3. When N = 1, the computational savings are a factor of 64, and the image quality may be acceptable for some applications. When N = 3, there is some computation savings, and the image quality is very close to that achieved with conventional decompression and color correction. The computational savings for different values of N are summarized in Table 11.5.

11.8 Color transforming multiresolution images

The previous section exploited the reduced sensitivity of the human visual system in high spatial frequencies to develop an efficient color transforma-

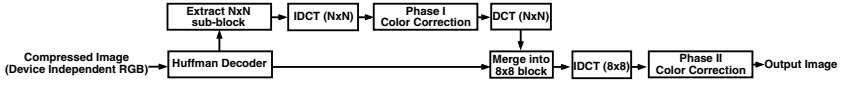


Figure 11.14 Efficient approach of combining color correction with JPEG decompression.

Figure 11.15 Wavelet representation of an image.

Table 11.5 Percentage Savings in Employing Efficient Correction of JPEG Compressed Images vs. the Standard Approach

Operation	N = 1	N = 2	N = 3
Lookups	40.9	37.2	29.7
Comparisons	18.8	13.8	6.3
Additions	48.8	43.3	33.6
Multiplications	98.8	95.3	89.5
Shifts	46.9	37.5	34.4

tion algorithm for JPEG imagery. The technique presented in this section is similarly motivated but applied to images represented in a multiresolution format.[10] In many applications, images undergo a multiresolution decomposition (e.g., discrete wavelet transform) for the purpose of image compression or other processing. After the image is reconstructed from its multiresolution representation, it typically must be color corrected for rendering to a specific device. As in the previous section, the color correction process is divided into two phases: (1) a complex multidimensional transform (Phase 1), and (2) a series of essentially one-dimensional transforms (Phase 2). Phase 1 correction is then moved within the multiresolution reconstruction process in such a way that a small subset of the image samples undergoes the multidimensional correction. Phase 2 correction is then applied to all image samples after the image is reconstructed to its full resolution.

To illustrate the idea, a particular example is considered wherein a discrete wavelet transform (DWT)[11] is used to represent the image in a multiresolution format for an application such as compression[12] or inverse

halftoning.[13] A four-colorant (CMYK) printer is used as the output device to which the final image is rendered. This technique uses the multiresolution image representation to perform accurate color correction on a low-resolution image and then to perform a partial correction to the full resolution image prior to printing. The prime advantage is reduced computational cost required by the color correction. Another potential advantage is to reduce the effect of high-frequency noise in the color correction. This technique differs from that described in the previous section in that it is not restricted to the JPEG framework, but it addresses a more general architecture wherein multiple resolutions of an image (equivalently a frequency band decomposition of the image) are available at some point in the image processing path.

11.8.1 Wavelet representation

Numerous methods exist to decompose images into multiple resolutions. We use the wavelet transform[11] as an example of a multiresolution technique, as this is a powerful and general framework that has shown considerable promise in several applications such as compression. The wavelet transform, assumed to operate independently on each of the color separations, is applied using a cascade of two-channel filter banks. These filter banks are composed of decimators (i.e., downsampling operators) and lowpass and highpass filters. At a given level, the image is split into four sub-bands (lowpass and highpass in each direction). The wavelet transform is obtained by repeating, at each level, the downsampling and filtering process over the lowpass sub-band of the previous level. This is shown in a pictorial example in Figure 11.15 for four levels. The image is then processed or analyzed in the wavelet domain for the given application and is finally submitted to an inverse wavelet transform so as to reconstruct the spatial domain representation. Note that since each downsampling or upsampling operation is accompanied by a filtering operation, the multiresolution representation can also be conceptualized as a decomposition of the image into a series of frequency bands, i.e., a sub-band decomposition. Such a decomposition trades frequency resolution for spatial resolution. The reader is referred to Chapter 8 for further details on the wavelet transform. The DWT is assumed to already be a part of the system for a given application such as compression (see Figure 11.16); we simply take advantage of its availability for efficient color correction.

11.8.2 Combining multiresolution analysis and color correction

The technique is implemented by moving Phase 1 color correction inside the inverse wavelet (i.e., image reconstruction) transform as shown in Figure 11.17. The image is partly reconstructed in the device-independent color space. Phase 1 color correction is then applied to convert the subsampled image to a device color space, and the remaining wavelet reconstruction is

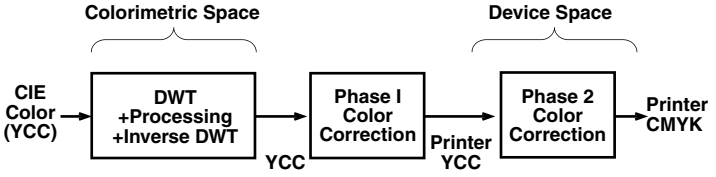


Figure 11.16 Standard approach of color correcting an image in the wavelet representation.

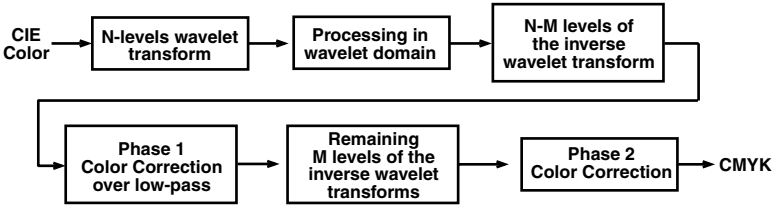


Figure 11.17 Efficient approach of performing color correction of a wavelet representation.

applied in this device space. Finally, Phase 2 correction is applied to the fully reconstructed (i.e., full resolution) image.

As with the JPEG case described above, the input color space to the Phase 1 transform must have the same sense and orientation as the output color space (e.g., colorimetric RGB to device R'G'B'). This is because the frequency bands that have gone through Phase 1 color correction are now combined with the remaining frequency bands that have not been color corrected, and such combinations make sense only if the image data in all the bands are of the same color sense.

Note that, when performing N stages on the forward transform over an image of $P \times P$ pixels and only $N - M$ stages in the inverse transform, the result is a subsampled image with dimensions $P' \times P'$, where $P = 2^M P'$. Hence, Phase 1 color correction is applied to only a small fraction $(1/2^{2M})$ of the samples, thus drastically reducing the computational cost. If C_1 and C_2 represent the computational cost per pixel of Phase 1 and Phase 2 color corrections, respectively, the overall cost for the proposed system is

$$C = C_1\, 2^{-2M} + C_2 \tag{11.22}$$

This is to be compared with the complexity of the standard color correction, which is on the order of $4C_1/3$ (since three-dimensional interpolation has to be performed for each of C, M, Y, K as opposed to three color separations in the proposed method). The wavelet transform is not included in the cost because it is assumed to be needed for other reasons.

Table 11.6 shows the computational cost per sample necessary to color correct the image using the low-cost and conventional approaches for M =

Table 11.6 Comparison of Computational Cost per Image Sample Using Standard and Proposed Multiresolution Color Correction Approaches

Operation	Standard	$M = 1$	$M = 2$	$M = 3$
Lookups	19	14.75	11.94	11.23
Comparisons	2.5	2.63	2.15	2.04
Additions	14	9.75	7.67	7.17
Multiplications	12	2.25	0.56	0.14
Shifts	2	1.25	1.06	1.02

1, 2, 3 (i.e., Phase 1 applied to 1/4, 1/16, and 1/64 of the samples, respectively). The savings provided by the proposed method is significant, particularly for multiplications, additions, and table lookups.

The parameter M offers the trade-off between computational cost and image quality. Higher values of M result in more computational savings but greater image degradation. Since the degradation is introduced at high spatial frequencies, where the HVS has reduced sensitivity, an acceptable quality/cost trade-off should be achievable with this technique.

Furthermore, because the color correction process itself is derived from measurements of targets of uniform patches, it is based on a low-frequency characterization of the device. Hence, there seems to be some consistency in applying the color correction to a low-frequency (i.e., locally averaged) version of the image. The high-frequency information must somehow be added back to the image to preserve texture and edges. However, because there is no explicit color characterization of the device at these frequencies, a simple approximation can be used to add the information in these bands, exploiting again the frequency response of the HVS.

11.8.3 Results

Balasubramanian et al.[10] have examined the cost/benefit trade-off for wavelet-based color correction. For this experiment, no application-dependent processing (e.g., compression, segmentation, etc.) in the wavelet domain was included. That is, the wavelet transform was computed up to a given stage M (where $M = 0$ represents the original image at full resolution); Phase 1 color correction was performed on the lowpass channel at that stage; the inverse wavelet transform was performed; and finally, Phase 2 color correction was applied on the entire image. A 16-tap QMF (Johnston) filter bank[11] was used for the wavelet analysis. The images were processed through the standard and wavelet-based techniques and reproduced on a Xerox 5760 CMYK xerographic printer at 400 dpi.

For $M = 1$ or $M = 2$, pictorial images corrected with the proposed algorithm result in quality that is very close to, if not indistinguishable from,

that of the standard approach. For $M = 3$, artifacts start to appear in the form of *beating* or *oscillations* in reasonably flat areas of the image. These artifacts are mainly noticeable in the luminance component of the image, so similar effects might also appear if one would apply the same concept for TRC adjustments of monochrome images. They result from the printing process's highly nonlinear nature and from mixing corrected and uncorrected frequency bands.

The algorithm's resulting image quality was also measured quantitatively. To provide a visually meaningful representation, the CMYK images were transformed into CIELAB space using a printer characterization function. To maintain consistency throughout the system, the same characterization function was used to derive the color correction transforms. However, while the colors are now in a perceptually meaningful representation, the CIELAB coordinate system suffers from the drawback that it is intended only for comparison of large, uniform patches and does not account for the sensitivity of the HVS to different spatial frequencies. This limits the utility of the CIELAB space for evaluating differences between reproductions of natural scenes. Since the algorithm exploits frequency characteristics of the HVS, an error metric should also be spatial frequency dependent. To this end, the spatial CIELAB (sCIELAB) model[14] was used. This model uses the luminance and chrominance spatial contrast sensitivity functions, as well as the traditional CIELAB color space, to derive a color difference metric that accounts for both spatial frequency characteristics and color difference sensitivity of the HVS. For the color difference term, the original sCIELAB model uses CIELAB 1976 ΔE. In this work, we used the CIE 1994 color difference equations as a superior metric for perceived color difference. The final sCIELAB error metric is in the form of an error image that represents the perceived difference between two images. An aggregate error criterion was selected as

$$\Delta E_{rms} = \sqrt{\frac{1}{S}\sum_{ij} \Delta E_{ij}^2} \tag{11.23}$$

where S is the number of image pixels, and ΔE_{ij} are the individual sCIELAB ΔE errors for each pixel. Tests were carried out for $M = 1$, $M = 2$, and $M = 3$, for three images with varying pictorial content. A viewing distance of 14 in. and a 400 dpi resolution were assumed. The average ΔE across the three images is shown as a function of M in Table 11.7. In general, $\Delta E < 1$ is considered to be below the visual detectability threshold. Noise, printer instability, and other imperfections in the system can easily give rise to reproduction errors between $\Delta E = 1$ and $\Delta E = 3$. Given this margin, and the qualitative observations, $M = 2$ appears to provide substantial savings with acceptable image quality. We remark, finally, that to make the evaluation more realistic, the effect of wavelet processing for the given application (e.g., compression, descreening, etc.) must also be considered.

Table 11.7 sCIELAB ΔE Error between Proposed Multiresolution Color Correction Scheme and the Standard Approach ($M = 0$)

M	0	1	2	3
sCEILAB ΔE_{avg}	0	1.29	1.84	3.19

Note: Printed image resolution is 400 dpi, and the assumed viewing distance is 14 in. Results are averaged over three images.

11.9 Color transformations using multilevel chrominance halftoning

11.9.1 Introduction

As with the previous two approaches, this technique is based on the observation that the HVS is less sensitive to errors in chrominance than errors in luminance at high spatial frequencies. This fact may be exploited for efficient LUT interpolation in the case where the input to the LUT is a luminance-chrominance color space. Consider the example of a printer color correction LUT from L*a*b* to CMYK. The aforementioned property of the HVS suggests that it is possible to introduce high-frequency distortions in the a* and b* channels in a manner that is not objectionable to a human observer. One such distortion is multilevel halftoning[15] along each of the a* and b* axes. Multilevel halftoning in three-dimensional LUTs has been described by Spaulding et al.[16] and Love et al.[17] In both cases, the authors apply halftoning to all three dimensions of the input signal, eliminating interpolation altogether. Here, we consider the special case of luminance-chrominance input and apply halftoning only to chrominance. By introducing high-frequency distortions only in the chrominance channels and preserving high accuracy in luminance, this technique offers image quality superior to that achieved by three-dimensional halftoning, at a modest computational cost.

Numerous multilevel halftoning techniques exist, e.g., screening, scalar error diffusion, vector error diffusion, etc. (These are covered in detail in Chapter 6.) Of these, multilevel screening is the least computationally intensive. The cost savings is achieved by choosing the halftone levels to coincide with the LUT node locations along a* and b*, as shown in Figure 11.18a. In this figure, the gray circle represents an input point in CIELAB space. First, multilevel halftoning is performed along the a* dimension by comparing the input a* value to a threshold in a spatial, periodically repeating halftone screen. If the input value exceeds the screen threshold, the former is mapped to the next larger level a^*_2 in the LUT; otherwise, it is mapped to the next smaller level a^*_1. This process is repeated for the b* dimension. Thus, after the chrominance halftoning step, the input color maps to one of the four neighboring LUT nodes in the a*-b* plane (denoted by black circles in Figure

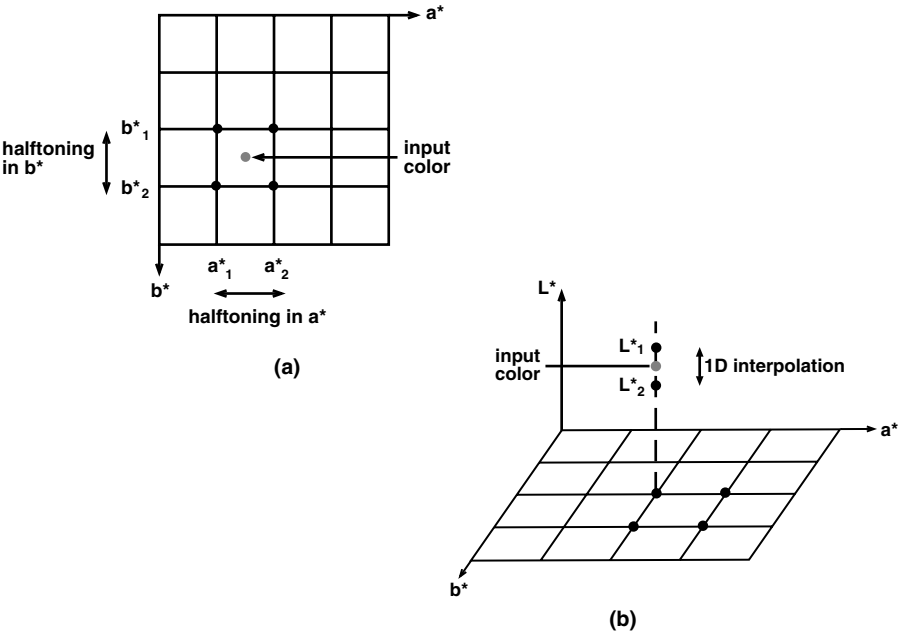


Figure 11.18 (a) Two-dimensional multilevel halftoning in the chrominance plane, followed by (b) 1-D interpolation in luminance.

11.18a). Calculating the output CMYK now requires only a one-dimensional interpolation along the L* dimension between the neighboring levels L^*_1 and L^*_2, as shown in Figure 11.18b.

The choice of halftone screen is an important consideration. For Balasubramanian's experiment,[1] an 8× 8 Bayer screen was used whose threshold values are shown in Table 11.8. (The threshold values as well as the input values are normalized to the range [0–1] prior to halftoning.) The consider-

Table 11.8 Threshold Values in Bayer Dot Screen Used for Multilevel Chrominance Halftoning

0	16	4	20	1	17	5	21
24	8	28	12	25	9	29	13
6	22	2	18	7	23	3	19
30	14	26	10	31	15	27	11
1	17	5	21	0	16	4	20
25	9	29	13	24	8	28	12
7	23	3	19	6	22	2	19
31	15	27	11	30	14	26	10

Note: Each of the values is divided by 31 before comparing with the input signal, assumed to be normalized to the range [0, 1].

able variation in threshold values between adjacent halftone cells gives the Bayer dot its ability to effectively contain the halftone signature within high spatial frequencies. Other screens, e.g., stochastic screens, can also be used. The same screen was used at the same spatial orientation for both a* and b* channels. It is conceivable that an improvement in image quality could be gained by using different screens or screen orientations for the a* and b* channels.

11.9.2 Results

To evaluate the quality of this technique, four test images of natural scenes were processed. These images, shown in grayscale in Figure 11.19, represent a variety of image content, including several smooth gradations, which are particularly challenging to multilevel halftoning. These images, specified in CIELAB, were transformed to CMYK with a three-dimensional LUT using the standard and halftone-based interpolation techniques. The LUT was constructed for a Xerox 5795 laser printer and comprised 16 uniformly spaced nodes along each of the three axes. Table 11.9 provides a comparison of the computational complexity of the novel chrominance halftoning technique with standard trilinear and tetrahedral interpolation. Recall that the latter is the fastest known three-dimensional interpolation algorithm. The table breaks down the computational cost in terms of multiplication, addition, comparison, and shift operations per pixel. The last column in the table shows actual execution times on a Sparc20 workstation. To account for variations from trial to trial, the timing data was derived by processing each of the four images through 10 trials and averaging the results across all the trials. Also, to make the comparison realistic, standard speed-up techniques such as caching were used with the trilinear and tetrahedral interpolation for the timing tests (these were not included in the computational cost measures). The data in Table 11.9 show that the cost savings are significant.

As in the previous sections, sCIELAB was used in conjunction with the CIE 1994 color difference metric to assess image quality objectively. The experiment was performed as follows. A three-dimensional LUT from L*a*b* to offset press CMYK was constructed using a standard printer

Table 11.9 Cost Analysis for Trilinear, Tetrahedral, and Chrominance Halftoning Methods, for *N* output Signals

Interpolation algorithm	M	A	C	S	T
1. Trilinear	7N	7N + 2	0	2	18.0
2. Tetrahedral	3N	3N + 2	2.5	2	12.0
3. Chrominance halftoning + luminance interpolation	N	N + 4	0	2	5.9
4. Percent savings from (2) to (3)	67	43	100	0	51

Note: M, A., C, and S denote multiplications, additions, comparisons, and shift operations, respectively. T denotes the real execution time in μs to perform the interpolation for each pixel on a Sparc20 workstation.

Figure 11.19 Images used in evaluating multilevel chrominance halftoning (clockwise from top left): Jill, Fruit, Carousel, Painted girl. "Painted girl" was obtained from the Kodak Photo Sampler (photographer: Steve Kelly).

characterization technique (see Chapter 5 for exemplary methods). The input L*a*b* image was transformed to printer CMYK with this three-dimensional LUT, using both the new technique and tetrahedral interpolation. The two CMYK images were then converted back to CIELAB using a printer characterization function for the offset printer and a visual difference image computed using sCIELAB. The mean and 95th percentile values from the difference image are reported in Table 11.10 for the four test images. Recall as a point of reference that, for an image consisting of a single color

Table 11.10 Average and 95th Percentile sCIELAB Errors for Images in Figure 11.19, Error Calculated between Reproductions Produced by Tetrahedral Interpolation vs. Chrominance Halftoning

Images	sCEILAB error	
	Average	95th percentile
Jill	2.59	4.4
Painted girl	3.38	6.0
Fruit	3.48	6.4
Carousel	4.0	8.1

(i.e., only the zero frequency component), the sCIELAB error reduces to CIE ΔE94. In our experience, the errors shown in Table 11.10 are within typical page-to-page and day-to-day print variations, which lie between 3 and 5 ΔE units for many printers. If LUT size is not a critical issue, a denser node sampling along a* and b* will further reduce the visibility of the chrominance halftoning.

Visually, it is usually difficult to distinguish the difference between chrominance halftoning and standard tetrahedral interpolation at normal viewing distances. Close examination sometimes reveals minor artifacts arising from the chrominance halftoning, particularly in smoothly varying image regions. On the whole, experimentation shows that the new technique works well for a wide variety of imagery, including computer generated graphics content. For printers with low spatial resolution, e.g., low-cost inkjet printers, the minor artifacts introduced by the chrominance halftoning are often masked by the final binary halftoning applied to each of the C, M, Y, K separations.

11.10 Conclusions

In this chapter, a number of techniques have been introduced to facilitate efficient color transformations. Since multidimensional LUTs pose the greatest challenges in terms of computational and storage costs, these have been the primary focus. The chapter began with an introduction to basic three-dimensional LUT interpolation geometries on regular lattices, including trilinear, prism, pyramidal, and tetrahedral. This was followed by a discussion of interpolation on irregular lattices, a special case being sequential interpolation. A set of practical techniques to accelerate interpolation operations was then described. Finally, various techniques were presented for reducing the computational cost of processing large images through three-dimensional LUTs. These techniques all exploit the human visual system's reduced sensitivity to luminance errors at high spatial frequencies. The most suitable technique is determined by the trade-offs between cost and accuracy relevant to a given application. High-speed printing systems, for example, may call

for techniques that reduce computation, while low-cost devices with limited built-in memory are likely to benefit from techniques that compress LUT size with minimal impact on quality. Hardware assistance can also be used to complement the techniques proposed in this chapter. Finally, conceivably, these techniques can be combined with each other to form even more effective approaches.

Acknowledgments

The authors would like to thank Dean Harrington for his assistance in preparing many of the figures for this chapter.

References

1. Balasubramanian, R., Reducing the cost of lookup table based color transformations, *J. Imaging Science & Technology*, 44(4), 321–327, 2000.
2. Hung, P-C., Colorimetric calibration in electronic imaging devices using a look-up table model and interpolations, *J. Electronic Imaging*, 2(1), 53–61, 1993.
3. Kang, H. R., *Color Technology for Electronic Imaging Devices*, SPIE, Bellingham, WA, 1997.
4. Edelsbrunner, N., Preparata, F., and West, D., Tetrahedrizing point sets in three dimensions, *J. Symbolic Computation*, 10, 335–347, 1990.
5. Allebach, J. P., Chang, J., and Bouman, C., Efficient implementation of nonlinear color transformations, in *Proc. IS&T and SID's 1st Color Imaging Conference: Transforms and Transportability of Color*, 143–148, 1993.
6. Chang, J. Z., Allebach, J. P., and Bouman, C. A., Sequential linear interpolation of multidimensional functions, *IEEE Trans. on Image Processing*, 6(9), 1231–1245, 1997.
7. Agar, A. U. and Allebach, J. P., A minimax method for sequential linear interpolation of nonlinear color transformations, *Proceedings of the Fourth IS&T/SID Color Imaging Conference*, Scottsdale, AZ, November 19–22, 1996, 1-5.
8. Aho, A., Sethi, R., and Ullman, J., *Compilers, Principles, Techniques and Tools*, Addison Wesley, Reading, MA, 433–438, 1986.
9. Klassen, R. V., Balasubramanian, R., and deQueiroz, R., Color correcting JPEG compressed images, in *Proc. IS&T and SID's 5th Color Imaging Conference*, 83–87, 1997.
10. Balasubramanian, R., deQueiroz, R., Fan, Z., Multiresolution color correction, in *Proc. SPIE*, 3300, 165–172, 1998.
11. Strang, G. and Nguyen, T. Q., *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, MA, 1996.
12. Said, A. and Pearlman, W., A new, fast, and efficient image CODEC based on set partitioning in hierarchical trees, *IEEE Trans. on Circuits and System for Video Technology*, 6, 243–250, 1996.
13. Luo, J., de Queiroz, R. L., and Fan, Z., Universal descreening technique via wavelet analysis, in *Proc. IS&T/SPIE Symp. on Electronic Imaging: Science and Technology*, San Jose, CA, SPIE 3018, 18–29, 1997.
14. Zhang, X. M. and Wandell, B. A., A spatial extension to CIELAB for digital color image reproduction, in *Proc. SID Symposium*, 27, 731, 1996.

15. Loce, R., Roetling, P. G., and Lin, Y-W., *Electronic Imaging Technology*, Chapter 7, SPIE, Bellingham, WA, 1999.
16. Spaulding, K. and Scott, K., Method and apparatus employing mean preserving spatial modulation for transforming a digital color image signal, U.S. Patent No. 5377041, issued December 27, 1994.
17. Love, S. T., Weed, S. F., Daniel, S. W., and M. E. Lhamon, Converting color values using stochastic interpolation, in *Proc. SPIE,* 3963, 208, 2000.

chapter twelve

Color image processing for digital cameras

Ken Parulski
Kevin Spaulding
Eastman Kodak Company

Contents

12.1 Introduction

Color digital cameras are used by a growing number of consumers and professional photographers. These cameras use one or more CCD or CMOS image sensors to capture color records of the scene, and they digitally process the color records to produce color image files. Digital cameras are normally just one part of an imaging system or imaging chain.[1]

The system typically includes an image-capable computer, which allows the digital camera images to be stored, edited, enhanced, printed, or transmitted to other locations.

Digital cameras are used in many diverse applications having varying requirements. These cameras include a wide range of designs, from "toy" PC peripherals used by children to color infrared cameras used in specialized scientific applications. Of these, two important classes of cameras can be compared: megapixel consumer cameras, which fully process or *render* the color image data and then store images in a standard JPEG-based image format, and professional cameras, which store a much higher quality *unrendered* representation of the color image data from the sensor. The consumer cameras produce "finished" JPEG standard image files that can be directly used by most imaging devices and software. The professional cameras store unrendered image data, which later will be processed using proprietary software running on a separate host computer to complete the camera image processing. This provides the highest possible quality and control of the color image.

Both types of cameras use a range of digital storage media, including several types of flash EPROM memory cards as well as magnetic and optical storage media. Professional cameras typically leverage conventional film camera bodies and removable lenses, and they use a large optical format CCD sensor.[2] Consumer cameras normally use a nonremovable zoom or fixed focal length lens designed to work with a small optical format sensor.[3]

12.2 Digital camera architecture

The capabilities and performance of a digital camera depend on both the camera's hardware architecture and its image processing algorithms, often provided by firmware. As a result, two cameras that share a common

hardware architecture and use identical image sensors can provide very different color image data if different image processing algorithms are used.

12.2.1 *Digital camera hardware*

Figure 12.1 is a generic block diagram of the hardware components used in a typical digital camera.[4] The camera has an adjustable focal length lens, controlled by zoom and focus motors. The zoom motor also controls an optical viewfinder. The lens assembly includes an infrared (IR) blocking filter and an optical anti-aliasing filter.[5]

The camera lens focuses light from the scene onto a single color image sensor, typically a charge-coupled device (CCD). The analog signal from the CCD is amplified and converted to digital form, normally using a 10- or 12-bit analog-to-digital (A/D) converter.[6]

The digital data are processed by a camera application-specific integrated circuit (ASIC) digital image processor[7] or by a high-performance microprocessor. Depending on the camera design, the camera image processing may be performed almost entirely in the camera ASIC or, alternatively, by a microprocessor incorporating a digital signal processor (DSP) designed to provide rapid image processing. In many cameras, the data from the A/D are temporarily stored in a DRAM buffer memory, and processing is performed on blocks of pixels. The DRAM buffer holds several unproc-

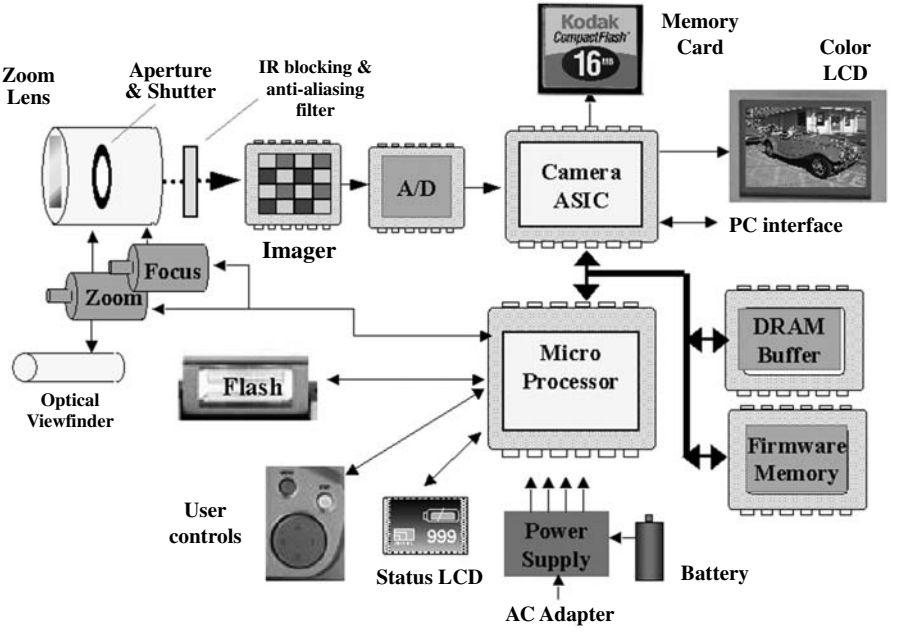


Figure 12.1 Block diagram of the hardware components used in a typical digital camera.

essed images, allowing *bursts* of images to be captured. The microprocessor is controlled by firmware stored in nonvolatile memory. In some cameras, the firmware is stored in flash EPROM and can be updated in the field to provide improved performance and new features.

The processed images are normally stored on a removable memory card, although some low-cost cameras use internal flash EPROM memory instead. There are several different memory card formats, including PC cards, CompactFlash cards, SmartMedia cards, Memory stick, SD (Secure Digital) cards, and multimedia cards (MMCs). Instead of flash EPROM cards, the removable memory can use magnetic hard drives, magnetic floppy disks, recordable optical discs, and magneto-optical discs. The removable media allow the images to be transferred to any computer or reader having the appropriate interface. The images can also be downloaded from the camera using a USB, IEEE 1394, or other common PC interface.

When the user presses the shutter button halfway, the camera performs automatic exposure and automatic focus processing to prepare to take the final still image. When the user fully depresses the shutter button, an image is captured and stored in DRAM. A subsampled image is processed to create a *thumbnail* size image for display on the LCD. This allows the user to immediately review the image and to capture a new image if the existing photo is unsatisfactory. The ability to view captured photos immediately is one of the most attractive features of digital cameras.

12.2.2 Color separation methods

There are several methods for capturing color images using a solid-state sensor, as shown in Figure 12.2 and described below.

- *Color filter arrays (CFAs).* Most digital cameras use a single image sensor overlaid with a mosaic pattern of colors, known as a *color filter array* or *CFA*.[8] Each photodetector is sensitive to only one color spectral band. As a result, *de-mosaicing* must be used to produce a full-color image. There are a variety of color filter patterns and de-mosaicing algorithms.
- *Color sequential.* In this method, the color image is produced by taking three successive exposures while switching in optical filters having the desired RGB transmission characteristics. The resulting color image is formed by combining the three-color separation images. The filters may be dichroic or absorptive RGB filters mounted in a color wheel or a tunable LCD filter. A few companies have developed professional color sequential digital cameras for studio use. These cameras are primarily used for still-life subjects, because any subject or camera motion will result in colored edges in the captured image.
- *Multi-sensor color.* This method uses a beam splitter, which is typically a dichroic prism, to separate the light into red, green, and blue

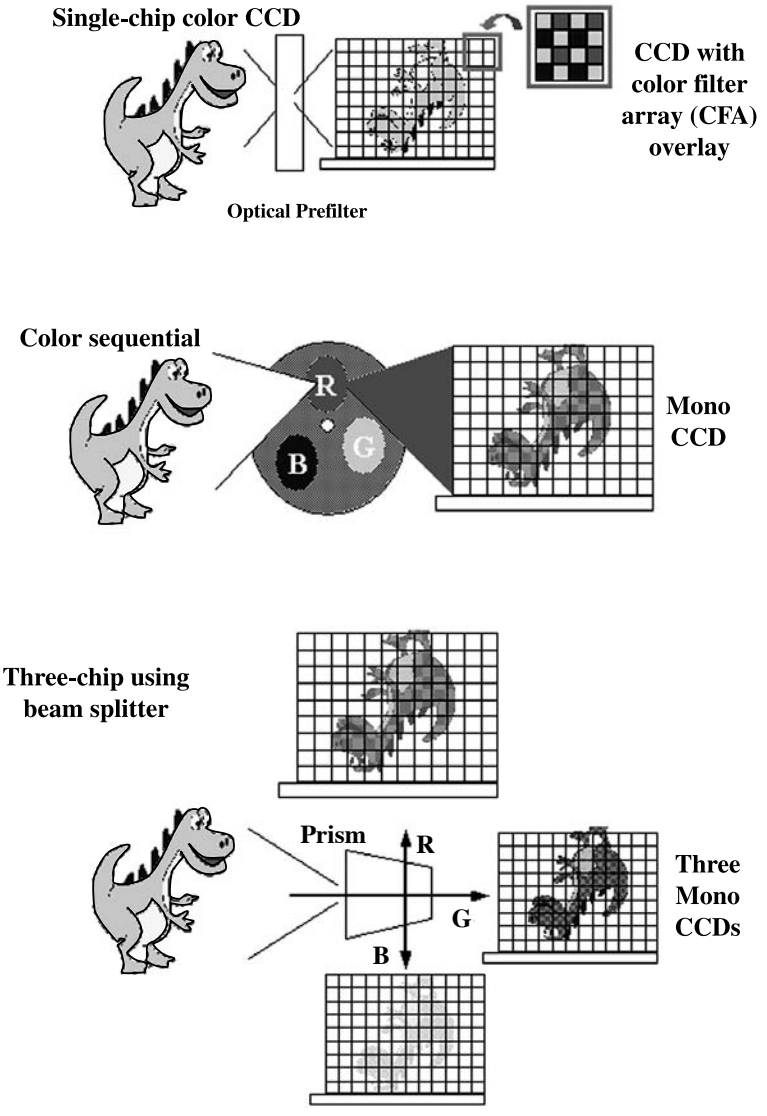


Figure 12.2 ***(See color insert following page 430)*** Capturing color images using a solid-state sensor.

components. These are focused onto three separate monochrome image sensors.[9]

A small number of professional cameras use this method. Cameras with a combined red/blue imager,[10] and one or two green imagers, have also been developed. The disadvantages of the multi-sensor approach include the high cost of the sensors and the beam splitter and the difficulties of maintaining image registration.

12.2.3 *Rendered camera processing*

To provide interoperability, most consumer digital cameras produce standard "rendered" image files that can be immediately used by most software applications. Figure 12.3 shows an example of the final still image processing flow in a typical consumer camera. An exposure and focus determination process provides the exposure time and the lens f/number and focus distance settings. The image is captured using these settings. The CFA data from the sensor is interpolated or "de-mosaiced" to reconstruct the "missing" color pixel values. White balance corrects for the scene illuminant, and a tone scale/color processing operation is used to compensate for the camera spectral sensitivities and render the image data. The rendered image data are sharpened, JPEG compressed, and stored using internal flash memory or a removable memory card. The images are stored using the Exif/JPEG format,[11] which requires that the images be represented as 24-bit RGB data and includes metadata indicating that the image code values should be interpreted as sRGB encoded color data.[12] This provides interoperability, but it limits the gamut and dynamic range of the stored images.

12.2.4 *Unrendered camera processing*

Digital cameras also have been developed for professional photographers, whose livelihood depends on taking high-quality color pictures. To provide the highest possible image quality, and to provide the flexibility needed to enable workflows appropriate for professional photography, the image processing can be divided between the camera and a separate host computer.

One example of a professional camera workflow is shown in Figure 12.4. The camera exposure/focus determination process provides the lens settings used to capture the color still image. The camera stores the image data captured by the sensor. To reduce the file size, an initial "rough" white balance can be used, followed by lossless image compression. The compressed sensor image data are formatted into a TIFF/EP image file[13] that

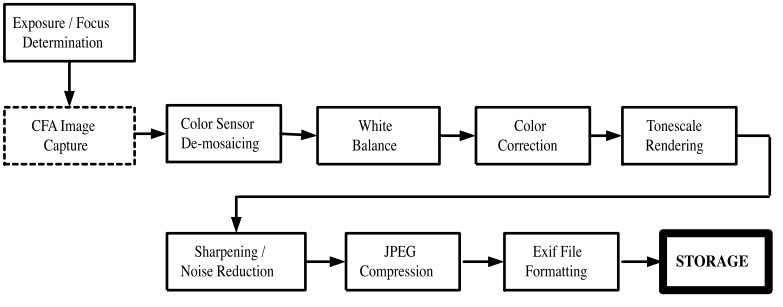


Figure 12.3 Example final still image processing flow in a consumer camera.

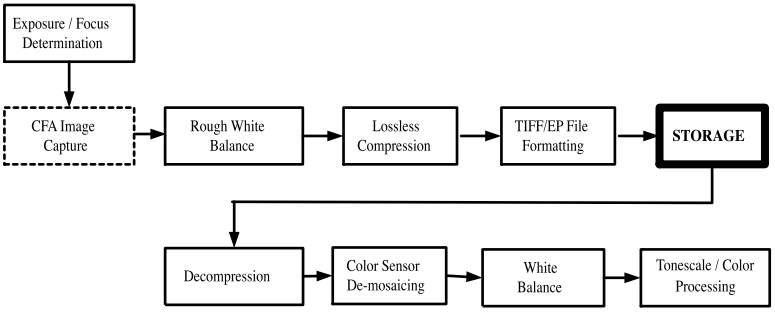


Figure 12.4 Example professional camera workflow.

includes metadata describing the camera characteristics, including the camera model and the sensor color filter pattern. The stored data are unrendered color sensor pixel values, typically called a *raw* or *unfinished* image file.

The images are downloaded to a host computer, which decompresses the image file to recover the image sensor color data. The de-mosaicing, white balance, and tone scale/color processing then can be performed. This can be done as part of an ICC color-managed workflow. Alternatively, the camera may provide some of these processing steps, such as de-mosaicing, before storing the unrendered image files.

The same digital camera hardware can provide both the rendered image processing shown in Figure 12.2 and the unrendered image processing workflow shown in Figure 12.3. For example, the KODAK PROFESSIONAL DCS 760 Digital Camera, a portable 6.3-megapixel camera, provides both types of processing. This camera uses a Nikon F5 35-mm format single-lens reflex (SLR) camera body and interchangeable lenses. The JPEG/Exif processing is typically used by photojournalists to immediately obtain finished images that can be transmitted via news agencies. The unrendered workflow is typically used by studio photographers to provide precise, high-quality color images with maximum flexibility.

12.3 Color image sensors

The core of a digital camera is a sensor array that converts light into electrical signals representing a color image. There are many different image sensor architectures[14] that can be used in digital cameras, including full-frame CCD sensors,[15] frame transfer (FT) devices,[16] interline (IL) devices,[17] frame–interline transfer (FIT) devices,[18] MOS x-y addressed devices,[19] and CMOS sensors.[20]

The primary difference among these sensor architectures is the method used to read the signal produced by the two-dimensional array of photodetectors so as to provide a sensor output signal. However, the type of photodetector used, rather than the charge readout architecture, is the most important color reproduction consideration, as determines the sensor's spectral

sensitivity. There are two classes of photodetectors (photodiodes and photocapacitors), each of which has different spectral response characteristics.

12.3.1 *Full-frame CCDs*

Many professional digital cameras use a full-frame CCD,[21] shown in Figure 12.5. This architecture consists of a two-dimensional array of cells, each having a light-sensitive photocapacitor. The photocapacitors are arranged as parallel, vertical charge-coupled device (CCD) shift registers. This enables the charge collected by the photocapacitors to be shifted into an opaque serial, *horizontal* CCD register at the bottom of the array and then serially transferred to an output amplifier. When the camera's mechanical shutter is opened, the camera lens focuses the image onto the photocapacitor array. A positive bias voltage is applied to form a depletion layer in the silicon beneath some of the polysilicon or indium tin oxide (ITO) gates of the photocapacitors, creating potential energy wells in each cell of the array. The incident light generates electron-hole pairs, and the electrons accumulate in the wells. Because the number of electrons collected is proportional to the incident illumination level, the sensor has a linear response curve, unlike photographic film. This means that the digital code value representing each pixel from the A/D converter, corresponding to each photosite on the CCD, is linearly related to the number of electrons collected by the photosite.

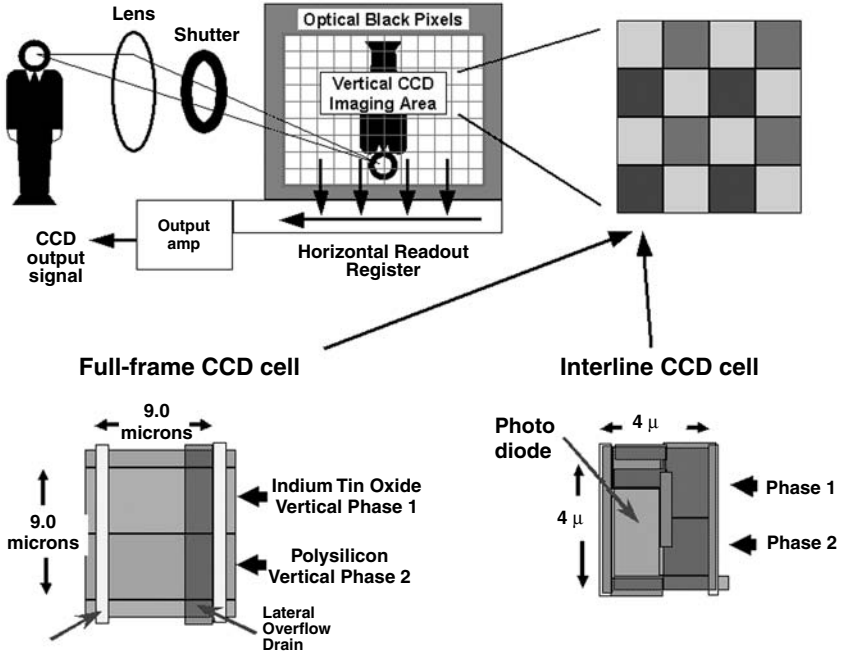


Figure 12.5 Full-frame CCD.

In very bright areas of an image (e.g., specular reflections), the number of electrons generated in a particular photodetector may exceed the charge capacity of the potential well. These excess electrons must be eliminated, or they will migrate to nearby wells, causing *blooming*, wherein the specular reflection grows into a blob or a streak. Image sensors used in digital cameras normally include an *anti-blooming* overflow drain to direct the excess charge into the sensor substrate and prevent it from spilling into neighboring cells. Nevertheless, setting the proper camera exposure level is critical, as overexposure will clip the image highlights. Underexposures can be corrected by digitally adjusting the code values of the final image, but this will also increase the noise.

At the end of the exposure time, the mechanical shutter closes to block the light, and the charge collected in each potential well is transported to the output amplifier. Clocking signals are used to repetitively cycle the voltages of adjacent electrodes in the array. The electrons are first shifted vertically, as a complete array, so that a row of electrons is transferred to the horizontal readout register. The electrodes in the horizontal readout register are rapidly clocked to transfer the electron charge packets, one at a time, onto a charge-to-voltage converter and output amplifier, which provides the sensor output signal.

The full-frame CCDs used in most professional digital cameras have cell sizes ranging from about 6×6 microns up to 16×16 microns. A larger cell size provides higher dynamic range and lower noise, enabling the camera to operate at a higher ISO speed. However, the larger cell size also makes the sensor physically larger and more costly, and the camera lens is likewise more bulky and expensive.

12.3.2 Interline CCDs

Most consumer megapixel cameras use an interline (IL) sensor architecture, also shown in Figure 12.5. Because they have a design similar to the sensors used in high-volume consumer camcorders, these CCDs can be manufactured at relatively low cost. The size of each cell is typically between 2.5×2.5 and 5×5 microns, so the sensor size is very small as compared to the size of sensors used in professional cameras. In most IL sensors, photodiodes located adjacent to each opaque CCD readout register are used as the light detectors. Whereas the photodiode uses only a small fraction of the cell area, a microlenticular array[22] is normally fabricated on top of the CCD to improve sensitivity. A microlens centered above each photodiode gathers much of the light that would otherwise fall on the opaque shift registers, and it directs the light into the photodiodes.

In most modern IL CCD sensors with small pixels, an interlaced readout method is employed, as the polysilicon gates of the vertical shift register cannot be made small enough. In this case, a vertical shift register cell is shared with two photodiodes. The odd and even rows of the photodiode array are read out separately. When IL sensors with interlaced readout are

used in a digital camera, a mechanical shutter is used to prevent additional accumulation of electrons in one set of photodiodes while the other set is being read out.

12.3.3 CMOS image sensors

Unlike CCDs, CMOS sensors can be fabricated using the same process technology used to make standard microprocessor and memory ICs. CMOS devices can integrate more functions — detection, readout control, and A/D conversion — onto a single integrated circuit. They consume far less power because of their higher level of integration.

In a typical CMOS sensor, each photosensitive cell contains a photodiode that converts light to electrons, a charge-to-voltage conversion section, a reset and select transistor, and an amplifier section. Overlaying the entire image-sensing array is a grid of metal interconnects to apply timing and readout signals, and output signal interconnects for each column. The column output signal is connected to a set of decode and readout electronics located outside of the light-sensitive array. This architecture allows the signals from the entire array, or from a subset of the array, to be read out by a simple addressing technique. This flexible readout, which allows a low-resolution image to be output quickly for camera exposure control and focusing, is not available with a CCD.

One of the biggest issues with CMOS sensors is the pixel-to-pixel and column-to-column fixed pattern noise caused by variability in the gain and offset of the readout transistors and per-column amplifiers. This noise limits the types of digital cameras that can use CMOS sensors. Whereas CCDs have lower noise levels than CMOS imagers, they can capture a larger scene dynamic range, i.e., the ratio between the darkest and brightest illumination levels that can be detected.

CMOS sensors were first used in two very different types of digital cameras. The first type is miniature, low-resolution cameras, such as cameras built into cellular telephones, which offer limited image quality. In this application, the low cost, low power consumption, and high level of integration offered by CMOS are very attractive. The second type is a professional digital camera for studio use. In this case, three large sensors are used in conjunction with significant digital image processing so as to compensate for the sensitivity and noise disadvantages of CMOS sensors.

12.3.4 Color filter array patterns

To provide a color image, each cell of the sensor array is covered with a transmissive filter of a particular color to form a color filter array (CFA). Although many different CFA patterns have been developed, the most popular patterns are the two shown in Figure 12.6. The RGB Bayer pattern has 50% green cells arranged in a checkerboard and alternating lines of red and blue cells. The complementary mosaic pattern[24] has equal proportions of magenta (M)-, green (G)-, yellow (Y)-, and cyan (C)-sensitive photosites arranged in magenta–green and yellow–cyan rows. The position of the

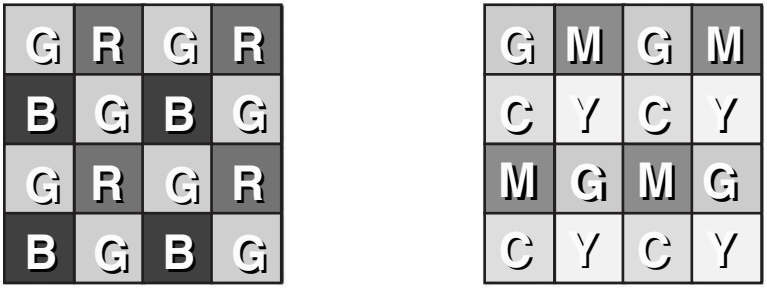


Figure 12.6 ***(See color insert)*** Popular CFA patterns.

green–magenta columns is staggered by one pixel on alternate green–magenta rows.

The color cells on the image sensor sample the color spectral bands of the scene at different spatial locations. As a result, high-spatial-frequency luminance information in the scene can be *aliased* by the color sensor sampling to produce low-spatial-frequency color artifacts. These artifacts can be extremely objectionable for certain types of scene details, such as regular patterns in architecture and certain textures in clothing. To minimize these aliasing artifacts, most digital cameras use some type of optical anti-aliasing, or *blur,* filter positioned in front of the sensor.[25]

People are more sensitive to high spatial frequencies in luminance than in chrominance, and luminance is composed primarily of green light. Therefore, the Bayer CFA improves the perceived sharpness of the digital image by allocating more spatial samples to the green image record. The checkerboard arrangement of the green cells in the Bayer CFA results in a diamond-shaped Nyquist domain for green, and smaller rectangular-shaped Nyquist domains for red and blue.[26]

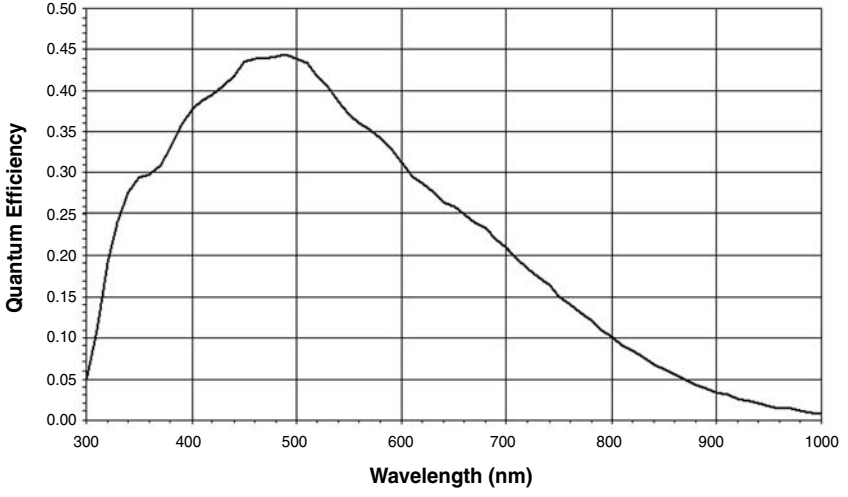
The RGB Nyquist domains are vertically and horizontally symmetric if the photosite pitch is *square* so that the limiting resolution due to the sensor sampling is the same in the vertical and horizontal directions.

The complementary pattern provides a higher sensor output signal level, because the cyan and yellow color filters absorb less than half the light of blue and red filters, for example. This provides an improved image signal-to-noise ratio at low illumination levels relative to RGB patterns, as the sensor output signal level is much higher. However, the RGB patterns normally provide a better image signal-to-noise ratio at higher illumination levels, when the sensor can operate at the full signal level, because the signal levels are equal, and the color correction processing required for complementary patterns increases the noise level.[27]

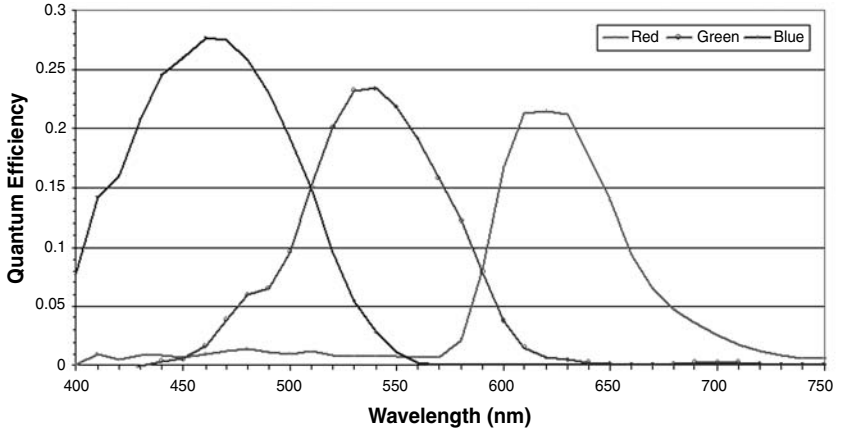
12.3.5 Sensor spectral response

Silicon semiconductors have an inherent sensitivity to light in the visible and the near-infrared regions of the spectrum. The quantum efficiency of a

typical photodiode used in an IL CCD is shown in Figure 12.7a. This is the response of a monochrome sensor having a microlens but no color filter array. The spectral response decreases above 500 nm because of the vertical overflow drain used to provide antiblooming protection. The long-wavelength photons penetrate deeper into the silicon before they are absorbed; therefore, the electrons they free have a higher probability of migrating to the antiblooming drain rather than to the potential well of the photodiode, as compared to shorter wavelength photons. Nevertheless, the IL CCD has



(a)



(b)

Figure 12.7 (a) Quantum efficiency of a typical photodiode and (b) color sensitivity of the output of a color IL image sensor.

significant sensitivity in the near-infrared region. Therefore, color digital cameras normally use an IR blocking filter to prevent color reproduction problems caused by IR light.

The color sensitivity of the output of a color IL image sensor in a typical digital camera is shown in Figure 12.7b. This figure shows the quantum efficiency of a typical IL sensor overlaid with an RGB pigment color filter array, cascaded with a typical camera lens and IR blocking filter. This color sensitivity will be compensated for by the camera's digital image processing, to produce a standard color response. Each camera can have a slightly different response because of the manufacturing tolerances of the sensor, filters, etc. In some cases, the compensation is determined specifically for each camera as it is manufactured so as to achieve the best results.

12.4 Color de-mosaicing in single-sensor cameras

To produce a high-quality color image, the color samples from a single-sensor color imager must be processed to provide red, green, and blue values (or luminance and chrominance values) at each pixel location, because each cell of the array captures only one spectral band. The process of filling in the missing color values has been referred to as color sensor *de-mosaicing*, *CFA interpolation*, and *color reconstruction*. A range of different types of algorithms, having various levels of complexity, can be used for different CFA patterns.[27]

Simple de-mosaicing algorithms use nonadaptive, bilinear interpolation. The missing color values are calculated using the average of the nearest samples of the appropriate color. For example, the missing green values (e.g., the green values at red and blue sensor cell locations) of the Bayer pattern are calculated by averaging the four values from the vertically and horizontally adjacent green cells. This approach is simple to implement but reduces the sharpness of the image while increasing the visibility of false color artifacts due to aliasing of high-spatial-frequency luminance patterns in the scene. Figure 12.8 shows an example of an edge-sensing de-mosaicing algorithm for calculating the missing green values.[28]

In this example, the missing green value, G5, is classified into one of three types: a uniform area, a horizontal edge, or a vertical edge. The classifier compares the absolute values of the differences of the vertically adjacent green pixels, G4–G6, and the horizontally adjacent green pixels, G2–G8, with a threshold value T. If both values are below the threshold, G5 is classified as being in a relatively uniform area and is set equal to the average of the four adjacent green values. This is equivalent to bilinear interpolation in uniform areas. If both values are not below the threshold, the absolute value of the difference between the horizontally adjacent green pixels, G4–G6, is compared with the threshold. If it exceeds the threshold, G5 is classified as a horizontal edge pixel and is set equal to the average of the two vertically adjacent green pixels. If not, G5 is a vertical edge pixel and is set to the average of the horizontally adjacent green values. This provides an adaptive algorithm, which interpolates *along the edge* to preserve edge

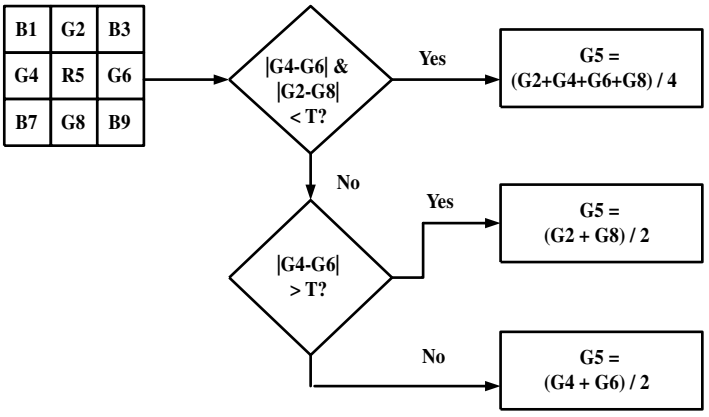


Figure 12.8 Example of an edge-sensing de-mosaicing algorithm for calculating the missing green values.

sharpness and minimize *zipper* artifacts that occur with simple bilinear interpolation.

Once the missing green values have been calculated, the luminance-like green record is complete, and the chrominance-like R/G and B/G records can be calculated. For example, whereas the values of B1/G1, G2, and B3/G3 are now known, the value of B2 is calculated by setting B2/G2 equal to the average of B1/G1 and B3/G3. In other words, the R/G and B/G values (rather than the R and B values) are calculated using bilinear interpolation. This approach calculates the missing red and blue values by effectively adding high-frequency luminance details from the green record into the sparsely sampled red and blue records.

More sophisticated de-mosaicing algorithms can provide sharper images, with reduced artifacts, by using a more complex classifiers[29] and by using a larger finite impulse response (FIR) filter[30] to calculate the missing green values. The de-mosaicing may instead use a pattern recognition approach (for example, by classifying the missing color pixels into edge, stripe, and corner patterns[31]) or a gradient-based approach[32] to adaptively interpolate the missing color pixel values. It can also be approached as an estimation problem that exploits knowledge of image spatial/spectral correlations.[33]

The color pixel reconstruction method used with the complementary mosaic pattern in Figure 12.6 normally recovers luminance (Y) and color difference signals R–Y and B–Y. Luminance can be obtained by lowpass filtering the color pixel values, because the magenta, green, yellow, and cyan filters have a luminance-like color spectral weighting. The color difference signals can be decoded by bandpass filtering the CCD output signal. This method reduces the sharpness of the luminance because of the lowpass filtering and can create false color artifacts. More elaborate algorithms exploit the correlation between the luminance and color difference signals to provide a sharper image with reduced false color artifacts.[24]

12.5 Exposure and white balance determination

Digital cameras must include an exposure determination system to properly expose the image sensor. Some digital cameras also adjust the effective exposure level of the captured digital image[34] (to compensate for sensor exposure errors) before storing the digital image file. The exposure of the image sensor depends on the lens f/number, sensor exposure time, scene illumination level, and scene reflectance, as well as many other secondary factors. Standard methods have been developed to measure the ISO speed rating of digital cameras to determine the optimal camera exposure.[35]

12.5.1 Exposure determination

Scenes contain a wide range of luminance values. To obtain an acceptable image, the sensor exposure for important areas of the captured scene must be less than the saturation level of the image sensor. If the sensor exposure level in a highlight area is too great, this portion of the digital image will be clipped. On the other hand, the sensor exposure should be large enough that the signal level in darker areas of the scene still provides an adequate signal-to-noise ratio.

The simplest camera exposure controls use a single photosensor to measure the average luminance in the lower center portion of the scene. The lower center weighting is used to reduce underexposure problems with many outdoor scenes that have bright sky in the top portion of the scene. The scene brightness value is read as the user begins to press the shutter button to take a picture. The brightness value determines the appropriate aperture and exposure time settings and, in some cases, the amount of signal amplification provided when reading out the image sensor.

More sophisticated methods use a multisegment light meter, or use the image sensor itself, to provide exposure control data. In the latter case, several images are taken in rapid succession, at different exposure levels, when the user depresses the camera shutter button halfway down.[36]

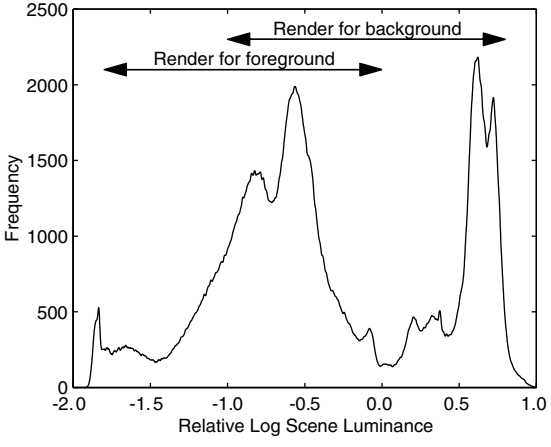
The resulting digital images can be divided into various segments, which are analyzed to determine the proper exposure level for the final still image, which is captured when the user fully depresses the shutter button.

12.5.2 Dynamic range

The proper exposure for a scene depends on the scene content and dynamic range. Figure 12.9 shows two images generated from a high-dynamic-range scene. The image on the left shows a rendering of the scene where the foreground information is properly exposed, and the image on the right shows a rendering of the scene where the background information is properly exposed. In the first case, much of the highlight information was clipped. Likewise, in the second case, much of the shadow information was lost. Also shown is a histogram of the scene luminance data for this image. A scene

(a)

(b)



(c)

Figure 12.9 ***(See color insert)*** Images generated from a high-dynamic-range scene.

luminance range of about 6 stops (1.8 log luminance units) can be reproduced within the dynamic range of a typical output medium. It can be seen that only a portion of the total scene information is reproduced in either of the images. As a result, it is critical that the original exposure level be chosen correctly so as to capture the information for the important part of the scene. Professional cameras that store the *unrendered* image data can typically retain a larger dynamic range than can be represented in a final rendered image. In this case, it is possible to make some adjustments in the exposure level when the image is processed on the host computer.

12.5.3 White balance determination

As a first step in providing good color reproduction, most digital cameras include a white balance operation. White balance requires adjusting the RGB signal levels provided by the image sensor to correct for the color temperature of the light source used to illuminate the scene. The amounts of red and blue light in daylight sources are approximately equal. However, many artificial light sources, such as tungsten light bulbs, provide a much higher proportion of red light than blue light. Images taken using these illuminants must have the blue signal amplified to prevent white objects from appearing unnaturally yellow in the reproduced image. This signal adjustment can occur in three different ways:

1. Using an optical color correction filter to equalize the sensor exposure levels
2. Increasing the analog gain when the low signal color values are read out of the sensor
3. Adjusting the digital code values of the captured image

Most digital cameras use the third method.

The difficult part of the white balance operation is determining which portions of the scene should be reproduced as neutral. The foolproof way to set the white balance is to have the camera user capture a white or gray card held at the scene, or to select a neutral object in the captured scene. This is appropriate for professional cameras but, for consumer applications, an automatic method for determining the illuminant color temperature is necessary.

One such method uses a set of photodiodes covered by RGB filters on the front or top of the camera, aimed upward toward the light source.[37] However, this approach can give incorrect results because of reflections from walls or other colored objects. It also requires separate components, which add cost.

Better results are normally obtained by estimating the proper white balance setting from the image sensor data. The captured color image is divided into segments, typically between 20 and several hundred, by averaging pixel values to create RGB values for each segment. The segment values are converted to color difference signals and analyzed to determine whether the segment is in the same region of the color difference space as a common illuminant, such as daylight, tungsten, and fluorescent regions.[38]

Figure 12.10 shows an example of a color difference space and the regions corresponding to various illuminants. The color difference values for the segments are analyzed, along with the absolute scene light level, to determine the likely scene illuminant. In some cameras, the averages of various segments are instead analyzed using fuzzy logic rules to determine the deviation from the white balance point.[39] Once the likely scene illuminant or the deviation from the white point is known, the correct separate RGB

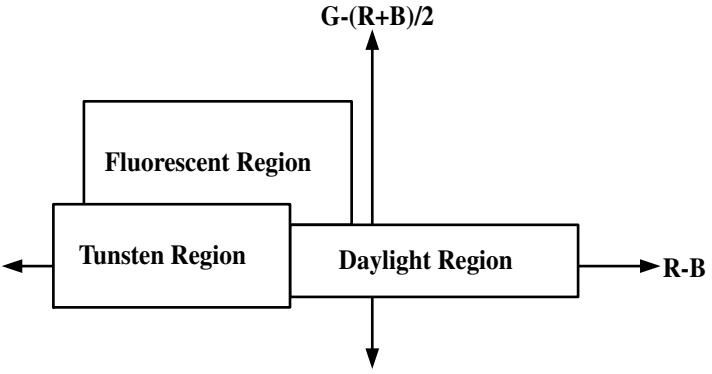


Figure 12.10 Example of a color difference space and the regions corresponding to various illuminants.

amplification factors can be applied to correct the white balance of the captured image.

12.6 Tone scale/color processing

The captured camera RGB image signals must be transformed to an output color encoding appropriate for display or printing on a selected output device. At a conceptual level, this process can be broken into three basic steps: a capture colorimetry model, a tone scale/color rendering step, and an output device model step. These steps are shown in Figure 12.11. While all digital cameras accomplish these operations either explicitly or implicitly, in many cases, there are no distinct boundaries between the steps. The purpose of each step will now be described in more detail.

12.6.1 Capture colorimetry model

The capture colorimetry model step relates the sensor RGB code values (after sensor de-mosaicing, in the case of a single-sensor camera) to an estimate of the colorimetry of the original scene. Generally, this model will be different for each different type of digital camera, reflecting the differences in the sensor's spectral sensitivities as well as any nonlinear function used to encode the sensor signal values. In some cases, it may even be desirable to

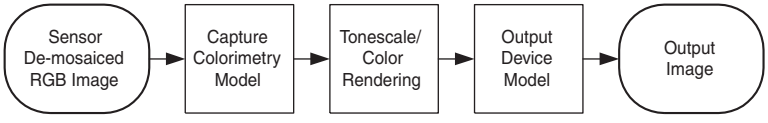


Figure 12.11 RGB image signal transformation process.

use a capture colorimetry model that is determined specifically for each individual camera.[40]

Generally, the capture colorimetry model will also be a function of the spectral characteristics of the scene illuminant. Because digital cameras are used to capture images under a variety of different illuminants, a different capture colorimetry model should ideally be used in each case. In most digital cameras, the only provision for compensating for different scene illuminants is the application of different channel-dependent gain values as part of the white-balance process, as described earlier. While this approach can be used to bring neutral scene objects back to the expected position, it cannot fully correct for other scene colors. It is possible for digital cameras to implement more complex changes in the camera model as a function of the scene illuminant.[41] In such cases, the appropriate camera model for the specific scene illumination conditions may be selected based on the results of the white balance determination described earlier.

Because color appearance is very much a function of the viewing conditions, the color appearance associated with the scene colorimetry can be known only when the conditions under which it is viewed are specified. Therefore, the estimated scene color values produced by the capture colorimetry model must be associated with a particular viewing environment to unambiguously represent a desired color appearance. In many cases, it is desirable to define a reference-viewing environment to which all images are referenced. This does not imply that the images need to be captured in that reference environment but rather that the camera model needs to correlate the camera signals to the color value with the desired color appearance in that reference-viewing environment. In practice, one way to specify the "desired appearance" is to define it as the appearance of the scene had it been viewed in the reference-viewing environment. Therefore, the capture colorimetry model can be built by capturing an image of a set of color patches under the expected actual illumination conditions and determining a transform to relate these values to the measured color values for the color patches in the reference-viewing environment. However, in other cases, it may be desirable for the color model to preserve the original look of the actual scene-viewing environment. For example, a photographer may desire that a photograph of a candlelit scene preserve the warm, desaturated look of the original scene rather than the look that the scene would have had under bright daylight illumination.

The output of the capture colorimetry model is an estimate of the scene colorimetry. However, there are many ways by which these scene color values could be encoded. For example, the color values could be represented using device-independent color spaces such as CIE XYZ tristimulus values or CIE L*a*b* (CIELAB) color values. Another alternative would be to use a scene-referred color encoding such as reference input medium metric RGB (RIMM RGB).[42]

This color encoding represents the scene colors in terms of a set of imaginary wide-gamut additive primaries. As will be described later, the

use of this color encoding has the advantage that desirable tone scale/color reproduction characteristics can be accomplished in the tone scale/color rendering step using simple one-dimensional lookup tables (LUTs).

The level of complexity used in the computations for the capture colorimetry model can vary widely from relatively simple matrix-based models to more sophisticated models requiring the use of complex transformations such as three-dimensional LUTs. Figure 12.12 shows one simple type of model that uses a LUT–matrix–LUT processing chain to compute RIMM RGB scene color values. The first set of one-dimensional LUTs is used to undo any nonlinearity that may have been applied to the sensor signals during the capture process. The output of these LUTs provides linear camera exposure values.

Next, a camera color-correction matrix is used to relate the camera exposure values to the scene tristimulus values. In theory, it can be shown that a 3×3 matrix operation can perfectly model this relationship, subject to the constraint that the sensor spectral sensitivities are expressible as linear combinations of the human visual system color matching functions. In practice, this condition is never exactly met, but a color-correction matrix can still be used as an approximate solution. Although an optimal color-correction matrix can be computed from the spectral sensitivities of the sensors, it is usually more convenient and more accurate to determine the matrix by applying a least-squares fitting technique to measured data for a set of color patches.

The following matrix can be used to compute the linear RIMM RGB values from the scene tristimulus values:

$$\begin{bmatrix} R_{RIMM} \\ G_{RIMM} \\ B_{RIMM} \end{bmatrix} = \begin{bmatrix} 1.3460 & -0.2556 & -0.0511 \\ -0.5446 & 1.5082 & 0.0205 \\ 0.0000 & 0.0000 & 1.2123 \end{bmatrix} \begin{bmatrix} X_{D50} \\ Y_{D50} \\ Z_{50} \end{bmatrix} \tag{12.1}$$

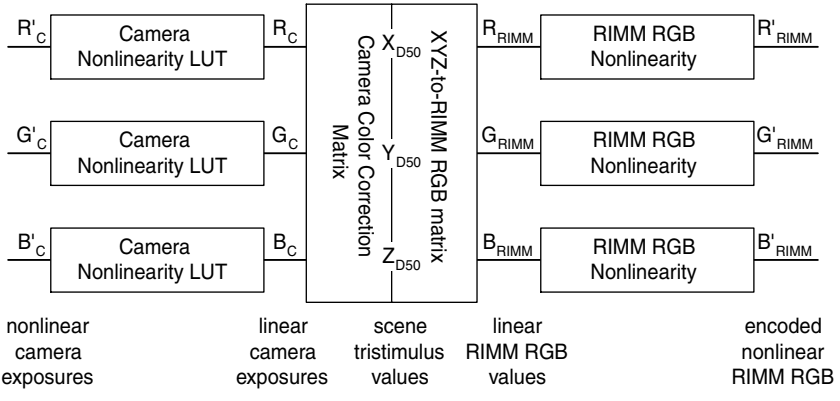


Figure 12.12 Model that uses a LUT–matrix–LUT processing chain to compute RIMM RGB scene color values.

(Note that this assumes that the camera color-correction matrix produces scene tristimulus values with respect to a D50 white point. If some other white point was assumed, a chromatic adaptation step would be required before applying this matrix.) Generally, the two matrix operations shown in Figure 12.12 can be combined into a single matrix multiplication for implementation purposes.

Finally, the linear RIMM RGB values are converted to an integer representation by applying the RIMM RGB nonlinearity.

$$C'_{RIMM} = \begin{cases} 0; & C_{RIMM} < 0.0 \\ \left(\frac{I_{\max}}{V_{clip}}\right) 4.5 C_{RIMM}; & 0.0 \le C_{RIMM} < 0.018 \\ \left(\frac{I_{\max}}{V_{clip}}\right) (1.099 C_{RIMM}^{\ 0.45} - 0.099); & 0.018 \le C_{RIMM} < E_{clip} \\ I_{\max} & C_{RIMM} \ge E_{clip} \end{cases} \tag{12.2}$$

where C = R, G, or B
I_{max} = maximum integer value used for the nonlinear encoding
E_{clip} = 2.0 is the exposure level that is mapped to I_{max}, and

$$V_{clip} = 1.099 E_{clip}^{\ 0.45} - 0.099 = 1.402 \tag{12.3}$$

12.6.2 *Tone scale/color rendering*

The next step shown in Figure 12.11 is tone scale/color rendering. This step is used to specify the desired relationship between the original scene colors and the rendered image colors. Many people would naively expect that the colorimetry of a pleasing rendered image would be an exact match of the colorimetry of the corresponding scene and that, therefore, this step should be a null operation. However, this is almost never the case. Among other things, the tone scale/color reproduction process that renders the colors of a scene to the desired colors of the output image must compensate for differences between the scene and rendered image viewing conditions.[43,44]

For example, rendered images generally are viewed at luminance levels much lower than those of typical outdoor scenes. Consequently, an increase in the overall contrast of the rendered image usually is required to compensate for perceived losses in reproduced luminance and chrominance contrast. Additional contrast increases in the darker regions of the image also are needed to compensate for viewing flare associated with rendered-image viewing conditions.

Psychological factors such as color memory and color preference also should be considered in image rendering. For example, observers generally remember colors as being of higher purity than they really were, and they typically prefer skies and grass to be more colorful than they were in the

original scene. The tone scale/color reproduction aims of well-designed imaging systems will account for such factors.

Finally, the tone scale/color reproduction process also must account for the fact that the dynamic range of a rendered image usually is substantially less than that of an original scene. It is, therefore, typically necessary to discard or compress some of the highlight and shadow information of the scene to fit within the dynamic range of the rendered image.

As with the scene-referred image encoding discussed earlier, there are many different color encodings that could be used to specify the desired color of the rendered image. For example, it could be represented by conventional CIE colorimetry or using the International Color Consortium Profile Connection Space (ICC PCS). (The ICC PCS is designed to serve as the connection between *input profiles* and *output profiles* for color management systems.) Another option is to use the reference output medium metric RGB (ROMM RGB) color encoding,[42] which is a companion color space to RIMM RGB. This color encoding uses the same set of wide gamut primaries as RIMM RGB but uses a different nonlinear encoding appropriate for the dynamic range of rendered images.

The complexity of the tone scale/color rendering transform can vary significantly, depending on the desired aims and implementation constraints. In some cases, complex tone scale/color reproduction aims may require the use of a three-dimensional LUT to independently control the color reproduction in different parts of color space. For many applications, generally desirable tone scale/color rendering characteristics can be achieved simply by applying a nonlinear tone scale LUT to the individual channels of the RIMM RGB image to form a rendered image in the ROMM RGB color encoding. (In fact, this was one of the design criteria that went into the definition of the RIMM RGB color encoding.[42]) A typical tone scale LUT of this type is shown in Figure 12.13.

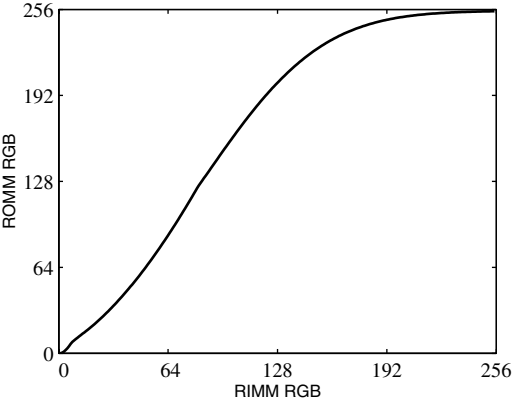


Figure 12.13 Typical ROMM RGB tone scale LUT.

12.6.3 *Output model*

The final operation shown in Figure 12.11 is the output device model. This step is used to determine the output code values necessary to produce the desired color on a given output device. For many workflows, the expected result is a video RGB image that is ready for direct display on a computer CRT. One particular video RGB color encoding, known as sRGB, has been adopted as an international standard[12] and is the default output color space for many digital cameras. One disadvantage to storing digital camera images in sRGB is that the color gamut of the image will be clipped to the gamut of the standard CRT. For some consumer applications, this will not be an important restriction. However, for color-critical, high-end applications, it is desirable to color manage images directly to an intended output device rather than going through an intermediate sRGB color encoding. This enables the highest possible image quality for those applications.

An example of an output model that can be used to convert a rendered image in ROMM RGB color encoding to sRGB is shown in Figure 12.14. First, linear ROMM RGB values are computed by applying the inverse nonlinearity,

$$C_{ROMM} = \begin{cases} \dfrac{C_{ROMM}}{16 I_{max}}; & 0.0 \le C_{ROMM} < 16 E_t \; I_{max} \\ \left(\dfrac{C'_{ROMM}}{I_{max}} \right)^{1.8}; & 16 E_t \; I_{max} \le C_{ROMM} \le I_{max} \end{cases} \tag{12.4}$$

where C_{ROMM} and C'_{ROMM} = nonlinear and linear ROMM RGB values, respectively
C = R, G, or B
I_{max} = 255 for 8-bit ROMM RGB

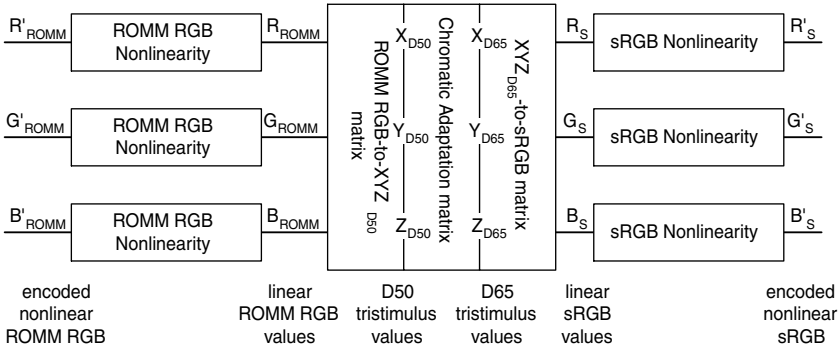


Figure 12.14 Example output model for converting an image in ROMM RGB color encoding to sRGB.

Next, the linear ROMM RGB values can be converted to D50 XYZ_{PCS} tristimulus values by applying the following matrix:

$$\begin{bmatrix} X_{PCS} \\ Y_{PCS} \\ Z_{PCS} \end{bmatrix} = \begin{bmatrix} 0.7977 & 0.1352 & 0.0313 \\ 0.2880 & 0.7119 & 0.0001 \\ 0.0000 & 0.0000 & 0.8249 \end{bmatrix} \begin{bmatrix} R_{ROMM} \\ G_{ROMM} \\ B_{ROMM} \end{bmatrix} \tag{12.5}$$

Whereas sRGB is defined using a D65 white point, a D50-to-D65 chromatic adaptation step must be applied before computing the sRGB color values. This can be accomplished using a simple von Kries transformation as follows:

$$\begin{bmatrix} X_{D65} \\ Y_{D65} \\ Z_{D65} \end{bmatrix} = \begin{bmatrix} 0.9845 & -0.0547 & 0.0678 \\ -0.0060 & 1.0048 & 0.0012 \\ 0.0000 & 0.0000 & 1.3200 \end{bmatrix} \begin{bmatrix} X_{D50} \\ Y_{D50} \\ Z_{D50} \end{bmatrix} \tag{12.6}$$

(The Hunt–Pointer–Estevez cone primaries[45] were used to derive this chromatic adaptation transform. Alternatively, other cone primaries or chromatic adaptation transforms could be used.)

The conversion from XYZ_{D65} tristimulus values to the linear *RGB* values associated with the sRGB primaries is given by the following inverse phosphor matrix:

$$\begin{bmatrix} R_S \\ G_S \\ B_S \end{bmatrix} = \begin{bmatrix} 3.2406 & -1.5372 & -0.4986 \\ -0.9689 & 1.8758 & 0.0415 \\ 0.0557 & -0.2040 & 1.0570 \end{bmatrix} \begin{bmatrix} X_{D65} \\ Y_{D65} \\ Z_{D65} \end{bmatrix} \tag{12.7}$$

Finally, the desired sRGB code values can be computed by applying the appropriate nonlinearity and quantization:

$$C'_S = \begin{cases} 255(12.92C_S); & C_S \le 0.0031308 \\ 255(1.055C_S^{1/2.4} - 0.055); & C_S > 0.0031308 \end{cases} \tag{12.8}$$

where $C = R$, G, or B.

For implementation, the three sequential matrix operations given in Equations 12.5 through 12.7 can be combined by cascading the matrices together to form the following single matrix:

$$\begin{bmatrix} R_S \\ G_S \\ B_S \end{bmatrix} = \begin{bmatrix} 2.0564 & -0.7932 & -0.2632 \\ -0.2118 & 1.2490 & -0.0372 \\ -0.0152 & -0.1405 & 1.1556 \end{bmatrix} \begin{bmatrix} R_{ROMM} \\ G_{ROMM} \\ B_{ROMM} \end{bmatrix} \tag{12.9}$$

Thus, the transformation from ROMM RGB to sRGB can be implemented with a simple LUT–matrix–LUT chain.

12.6.4 Processing configurations

The tone scale/color processing chain shown in Figure 12.11 is defined in terms of a sequence of modular operations. This provides significant flexibility in configuring the processing applied in a particular application. For example, different tone scale/color rendering modules can be used to achieve different looks according to different customer preferences, or different output models can be used according to the desired output device. There is additional flexibility as to when and where each of the processing steps is applied. For example, in the professional camera image processing workflow shown in Figure 12.4, the tone scale/color rendering and the output model steps are delayed so that the user can specify a preferred look and intended output device using a host-based image processing application. In other applications, such as the consumer camera image processing workflow in Figure 12.3, the final sRGB images are created and stored right in the camera. In part, this is because the current JPEG-based image format does not support alternative color spaces.

In many cases, it is desirable to cascade the various steps in the tone scale/color processing chain together to maximize computation efficiency. This is particularly true for the image processing workflow in Figure 12.3, where fast image processing is a primary consideration. For example, consider the processing steps shown in Figure 12.14. Because sequential matrix operations can easily be combined to form a single matrix, and, likewise, sequential LUT operations can be combined to form a single LUT, this entire processing chain can be implemented using a LUT–matrix–LUT–matrix–LUT operation.

To further reduce the complexity of the tone scale/color processing operations, the tone scale function can be moved to a position in the imaging chain following the sRGB conversion matrix. This effectively means that the tone scale is applied in video RGB primary space, rather than the wide-gamut primaries associated with the RIMM RGB space. This will result in some changes in the color reproduction characteristics of the image. The most significant changes will be in the hues of some saturated colors. For many applications, these differences will not be objectionable. After moving the tone scale LUT, several of the other LUTs and matrices cancel out, leaving the configuration shown in Figure 12.15. (It should be noted that the tone scale LUT shown in this figure is not the same one as that shown in Figure 12.13, because the input is now a linear exposure signal rather than encoded

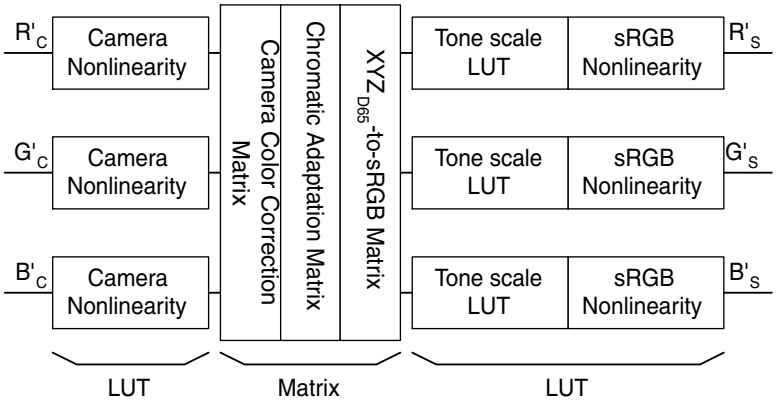


Figure 12.15 LUT-matrix-LUT sequence.

RIMM RGB as before. However, the same tone reproduction can be achieved by populating the LUT appropriately.) Because the groups of LUTs and matrices can be cascaded together, it can be seen that this imaging chain reduces to a simple LUT–matrix–LUT sequence. This is about as simple as the tone scale/color processing can get, and it is representative of the processing used in many consumer digital cameras.

12.7 Noise reduction and sharpening

The camera digital image processing can include several other types of processing, such as noise reduction and sharpening. Noise reduction can be performed at several different stages in the image processing chain. While sharpening is often performed immediately after tone scale and color processing, it can also be performed at an earlier stage of the image processing chain; for example, as part of the CFA de-mosaicing processing.

12.7.1 Noise reduction

While there are many sources of noise in a digital camera, the most significant is often the dark current variability of the image sensor. This noise is caused by minute imperfections or impurities in the sensor structure that generate a spatially and temporally varying number of "noise" electrons in each sensor cell, independent of the sensor illumination level. The signal from each cell includes electrons resulting from incident photons of light and electrons caused by dark current. The number of dark current electrons is proportional to the photosite integration and readout time and is an exponential function of the sensor temperature. Therefore, some professional cameras have used thermoelectric coolers to reduce the sensor temperature so as to reduce the dark current noise level.

In most digital cameras, the average dark current level is compensated by using "optical black" masked photodetectors at the edges of the image sensor. The average signal level of these black cells is determined and subtracted from each image pixel value. This ensures that the signal level of dark objects does not change with varying sensor temperatures or drifting dc circuit levels. In some professional digital cameras, the dark level and sensitivity variations of each pixel are determined as part of the camera manufacturing process and stored in the camera firmware. These pixel-by-pixel corrections are then performed on the digital image before de-mosaicing.

The color correction step can increase the amount of noise in an image. As a result, the correction factors may be modified to reduce the noise in the final image, at some sacrifice in color reproduction. Alternatively, the color correction can be performed for only the low-frequency image content.

12.7.2 Edge sharpening

The processing used in digital cameras often includes edge sharpening. This is done to compensate for the image blurring caused by the lens, optical anti-aliasing filter, and CCD aperture and to provide a subjectively sharper image. The appropriate amount of sharpening depends on the size and distance at which the final image is viewed. An image that appears "crisp" when printed at a small size may appear over-enhanced and artificial when enlarged.

Edge enhancement can be implemented using a two-dimensional convolution filter kernel, for example,

0	–1/4	0
–1/4	1	–1/4
0	–1/4	0

In this case, the filter kernel output equals the present luminance or pixel value minus the average of the four vertically and horizontally adjacent values. The filter output is zero in uniform areas of the image but non-zero for edges. More elaborate filter kernels can be used to tailor the frequency response. The filter output may be *cored* by setting small output values to zero, as these small values are often the result of noise. The signal is amplified and added back to the original luminance signal — or to the red, green, and blue signals — to increase the image sharpness.

Conventional wisdom would indicate that it is best to perform sharpening operations in a color space that is linear with exposure. This is because many of the sharpness degradations, such as optical blur, occur in linear space. However, in many cases, it has been found to be preferable to apply sharpening operations in a color space that is uniform with human perception so that the amount of sharpening is approximately independent of the color value. For example, a gamma-space encoding is much more visually uniform than a linear intensity space.

12.7.3 Chroma subsampling

Prior to image compression, the sRGB image signals are normally converted to luminance (Y) and color difference (R–Y, B–Y) signals. The color difference signals are often lowpass filtered and subsampled by a factor of 2:1 in both the vertical and horizontal direction (to create a 4:1:1 format image with four Y samples for each R–Y and each B–Y sample) or only in the horizontal direction (to create a 4:2:2 format image with four Y samples for every two R–Y and B–Y samples). To simplify the processing, the edge sharpening and noise reduction processing is often performed using the Y and subsampled R–Y and B–Y signals.

12.8 Image compression and file formats

Most digital cameras employ image compression to allow more images to be stored in the camera's digital memory. Consumer cameras normally use standard JPEG image compression, which is described in more detail in Chapter 8. Storing the JPEG compressed image data in a standard image file format enables the image to be decompressed and used by standard computer software applications such as Internet browsers and word processing programs as well as by image editing programs.

In professional cameras that require further host processing, it is possible to use a compression algorithm that is optimized for data from one-chip color sensors.[46] This can provide more efficient compression, as it operates on the sensor image data prior to de-mosaicing. Often, professional cameras will provide a lossless compression option to provide the highest possible quality at the cost of a larger file size.

12.8.1 Exif/JPEG image format

To transfer digital camera images to other digital devices, such as a computer, appliance printer, or photo kiosk, a standard image data format must be used so that the device can make sense of the data provided by the digital camera. Thus, standard image formats are even more important than standards for the physical and electrical compatibility of the transfer media.

Most megapixel consumer cameras use the Exif[11] image format, which is based on the JPEG standard.[47] The Exif files are named and organized into folders using the DCF standard.[48] DCF requires that the color image be interpreted as sRGB image data to ensure interoperability. In addition to the JPEG compressed image data, the Exif file stores various *metadata* describing the camera and picture-taking conditions.[49] The metadata include the time and date, camera zoom position and focus distance, illumination level, camera calibration data, subject, and copyright owner. The metadata can be used to simplify image retrieval and provide higher-quality prints from the digital files.[50] The Exif metadata include a thumbnail-size image at the beginning of the JPEG file to allow groups of images to be rapidly viewed

so that appropriate images can be quickly selected for viewing, copying, or printing.

12.8.2 TIFF/EP image format

Some professional cameras use the TIFF/EP[51] format, based on the TIFF revision 6.0 specification. Unlike JPEG-based formats, this format allows uncompressed or lossless compressed images to be stored. It also can be used with a variety of scene-referred or rendered color encodings such as RIMM RGB and ROMM RGB. However, to properly interpret the color data, the meaning of the digital color values stored in the digital image file must be identified and understood. This can be done by using an ICC color profile[52] to unambiguously define the RGB reference primaries, white point, and optoelectronic conversion function.

12.8.3 JPEG2000 image format

In the near future, digital cameras are likely to use a new wavelet-based compression standard known as JPEG2000, which has been approved as an International Standard[53] and is also described in Chapter 8. The JPEG2000 format provides the color space flexibility needed for future digital photography systems as well as greatly improved metadata support as compared with the old JPEG standard.

Many of today's digital cameras are capable of capturing color data with a wider dynamic range and larger color gamut than sRGB provides, but they have to discard the additional data to convert it into a JPEG file that is compatible with equipment from other vendors. With JPEG2000, digital cameras will be able to save an image using an extended gamut color RGB encoding, such as RIMM RGB or ROMM RGB, along with a restricted ICC profile that essentially provides instructions to allow the color image data to be converted into *sRGB* if needed. The extended range image can be used to create higher-quality prints and to allow exposure level and color adjustments of the decompressed images.

References

1. Whiteley, T. E. et al., Synergism: photography into the 21st century, *J. Soc. Photogr. Sci. Technol. Jpn.*, 53(2), 95–105, 1990.
2. Noble, S. et al., Design of high resolution digital cameras for pre-press applications, *TAGA Conf.*, Quebec City, May 1997.
3. Dunsmore, C. et al., A low-cost megapixel digital camera using high-performance in-camera image processing, in *Proc. IS&T PICS Conference*, 1998, 67–70.
4. Parulski, K. et al., The continuing evolution of digital cameras and digital photography systems, in *Proc. IEEE Int. Symp. on Circuits and Systems*, Geneva, Switzerland, May 2000.
5. Ogawa, K. et al., Development of CCD imaging block for single chip color camera, *IEEE Trans. Consumer Electronics*, CE-31(3), 405–412, 1985.

6. Koen, M., An analog-to-digital processor for camcorders and digital still cameras, *IEEE Trans. Consumer Electronics,* 44(3), 570–580, 1998.
7. Nakano, N. et al., Digital still camera system for megapixel CCD, I*EEE Trans. Consumer Electronics*, 44(3), 581–586, 1998.
8. Dillon, P. et al., Fabrication and performance of color filter arrays for solid-state imagers, *IEEE Trans. Electron. Devices,* ED-25(2), 97–101, 1978.
9. Edwards, E. et al., New digital photo-studio camera using two-directional spatial offset-type imager, in *Proc. IS&T's 47th Annual Conf.,* 658–660.
10. Okano, Y. et al., Electronic digital still camera using 3-CCD image sensors, in *Proc. IS&T's 48th Annual Conf.,* 428–432, 1995.
11. Digital Still Camera Image File Format Standard (Exchangeable Image File Format for Digital Still Camera: Exif, version 2.1), *JEID*A-49,1998.
12. IEC 61966 2-1:1999, *Multimedia systems and equipment — Colour measurement and management — Part 2–1: Colour management — Default RGB colour space sRGB.*
13. ISO 12234-2:2001 *Electronic still picture imaging — Removable memory, Part 2: Image data format–TIFF/EP.*
14. Sadashige, K., An overview of solid-state sensor technology, *SMPTE J.*, February, 180–186, 1987.
15. Miller, W. et al., A family of full-frame image sensors for electronic still photography, in *Proc. IS&T's 47th Annual Conf.,* 1994, 649–651.
16. Theuwissen, A., *Solid-State Imaging with Charge-Coupled Devices,* Kluwer Academic, Dordrecht, The Netherlands, 1995.
17. Steffe, W. et al., A high performance 190×244 CCD area image sensor array, in *Proc. Int. Conf. on Applications of Charge-Coupled Devices,* 1975, 101–108
18. Thorpe, L. et al, New advances in CCD imaging, *SMPTE J.*, May, 378–387, 1988.
19. Aoki, M. et al., 2/3-inch format MOS single-chip color imager, *IEEE Trans. Electron Devices,* 29(4), 745–750, 1982.
20. Fossum, E., CMOS image sensors: electronic camera on a chip, in *IEEE Int. Electron. Devices Meeting Tech. Digest,* December 1995, 1–9.
21. Chang, W. et al., High-density solid-state image sensor, *SMPTE J.,* December, 1186–1188, 1987.
22. Weiss, A. et al., Microlenticular arrays for solid state imagers, *J. Electrochem. Soc.,* 133(3), 110C, 1986.
23. Dillon, P. et al., Color imaging system using a single CCD area array, *IEEE Trans. Electron Devices,* 25(2), 102–107, 1978.
24. Sigiura, H. et al., False color signal reduction method for single-chip color video cameras, *IEEE Trans. Consumer Electronics*, 40(2), 100–106, 1994.
25. Grievenkamp, J., Color dependent optical prefilter for the suppression of aliasing artifacts, *Appl. Opt.*, 29(5), 676–684, 1990.
26. Palum, R., Image sampling with the Bayer color filter array, in *Proc. IS&T PICS Conf.,* 2001, 239–245.
27. Parulski, K. A., Color filters and processing alternatives for one-chip cameras, *IEEE Trans. Electron. Devices,* ED-32(8), 1381–1389, 1985.
28. Adams, J., Interactions between color plane interpolation and other image processing functions in electronic photography, *SPIE Proc.,* 2416, 144–151, 1995.
29. Kuno, T. et al., New interpolation method using discriminated color correlation for digital still cameras, *IEEE Trans. Consumer Electronics*, 45(1), 259–267, 1999.

30. Adams, J., Design of practical color filter array interpolation algorithms for digital cameras, *SPIE Proc.,* 3028, 117–125.
31. Cok, D., Reconstruction of CCD images using template matching, in *Proc. IS&T's 47th Annual Conf.,* 1994, 380–384.
32. Shimizu, E. et al., A digital camera using a new compression and interpolation algorithm, *Proc. IS&T's 49th Annual Conference,* 1996.
33. Taubman, D., Generalized Wiener reconstruction of images from colour sensor data using a scale invariant prior, *Proc. Int. Conf. on Image Processing,* Vol. III, 2000, 801–804.
34. Bhukhanwala, S. et. al., Automated global enhancement of digitized photographs, I*EEE Trans. Consumer Electronics*, 40(1), 1–10, 1994.
35. ISO 12232: 1998, Photography — *Electronic still picture cameras — Determination of ISO speed.*
36. Kuno, T., A new automatic exposure system for digital still cameras, *IEEE Trans. Consumer Electronics,* 44(1), 192–199, 1998.
37. Hanma, K. et al., Novel technologies for automatically focusing and white balancing of solid state color video camera, *IEEE Trans. Consumer Electronics,* CE-29(3), 376–382, 1983.
38. Mormura, A. et al., A digital video camera system, *IEEE Trans. Consumer Electronics,* 36(4), 866–876, 1990.
39. Liu, Y. et al., Automatic white balance for digital still camera, *IEEE Trans. Consumer Electronics,* 41(3), 460–466, 1995.
40. Vogel, R. et al., Digital Imaging Device Optimized for Color Performance, U.S. Patent No. 5,668,596, September 16, 1997.
41. Spaulding, K. E. et al., Method and Apparatus for Color Correcting Multichannel Signals of a Digital Camera, U.S. Patent, No. 5,805,213, September 8, 1998.
42. Spaulding, K. E. et al., Reference input/output medium metric RGB color encodings (RIMM/ROMM RGB), in *Proc. IS&T's 2000 PICS Conf.,* 155–163.
43. Hunt, R. W. G., *The Reproduction of Colour*, 5th ed., Fountain Press, England, 1995.
44. Giorgianni, E. J. and Madden, T. E., *Digital Color Management: Encoding Solutions,* Addison-Wesley, Reading, MA, 1998.
45. Fairchild, M. D., *Color Appearance Models,* Addison Wesley, Reading, MA, 1998, 250.
46. Tsai, Y. T., Color image compression for single-chip cameras, *IEEE Trans. Electron Devices,* 38(5), 1226–1232, 1991.
47. Pennebaker, W. B. et al., *JPEG Still Image Data Compression Standard*, Van Nostrand Reinhold, New York, 1993
48. JEIDA, Design Rule for Camera File System, Version 1.0, December 1998.
49. Watanabe, M. et al., An image data file format realized in PC card for digital still camera, *Proc. Int. Symp. on Electronic Photography (ISEP),* Koln, September 1994.
50. Milch, J. and Parulski, K., Using metadata to simplify digital photography, in *Proc. IS&T PICS Conf.,* 1999, 26–30.
51. Parulski, K. et al., TIFF/EP, a flexible image format for electronic still cameras, in *Proc. IS&T's 48th Annual Conf.,* 1995.
52. ICC.1:2001, *File Format for Colour Profiles*.
53. Christopoulos, C. et al., The JPEG2000 still image coding system: an overview, *IEEE Trans. Consumer Electronics*, 46(4), 1103–1128, 2000.