Budagavi, M. & Talluri, R. “Wireless Video Communications”
Mobile Communications Handbook
Ed. Suthan S. Suthersan
Boca Raton: CRC Press LLC, 1999

Wireless Video Communications

Madhukar Budagavi
Texas Instruments

Raj Talluri
Texas Instruments

31.1 Introduction

Recent advances in technology have resulted in a rapid growth in mobile communications. With this explosive growth, the need for reliable transmission of mixed media information—audio, video, text, graphics, and speech data—over wireless links is becoming an increasingly important application requirement. The bandwidth requirements of raw video data are very high (a 176×144 pixels, color video sequence requires over 8 Mb/s). Since the amount of bandwidth available on current wireless channels is limited, the video data has to be compressed before it can be transmitted on the wireless channel. The techniques used for video compression typically utilize predictive coding schemes to remove redundancy in the video signal. They also employ variable length coding schemes, such as Huffman codes, to achieve further compression.

The wireless channel is a noisy fading channel characterized by long bursts of errors [8]. When compressed video data is transmitted over wireless channels, the effect of channel errors on the video can be severe. The variable length coding schemes make the compressed bitstream sensitive to channel errors. As a result, the video decoder that is decoding the corrupted video bitstream can easily lose synchronization with the encoder. Predictive coding techniques, such as **block motion compensation,** which are used in current video compression standards, make the matter worse by quickly propagating the effects of channel errors across the video sequence and rapidly degrading the video quality. This may render the video sequence totally unusable.

Error control coding [5], in the form of **Forward Error Correction (FEC)** and/or **Automatic Repeat reQuest (ARQ),** is usually employed on wireless channels to improve the channel conditions. FEC techniques prove to be quite effective against random bit errors, but their performance is usually not adequate against longer duration burst errors. FEC techniques also come with an increased overhead in terms of the overall bitstream size; hence, some of the coding efficiency gains achieved by video compression are lost. ARQ techniques typically increase the delay and, therefore, might not be suitable for real-time videoconferencing. Thus, in practical video communication schemes, error control coding is typically used only to provide a certain level of error protection to the compressed video bitstream, and it becomes necessary for the video coder to accept some level of errors in the video bitstream. Error-resilience tools are introduced in the video codec to handle these residual errors that remain after error correction.

The emphasis in this chapter is on discussing relevant international standards that are making wireless video communications possible. We will concentrate on both the error control and source coding aspects of the problem. In the next section, we give an overview of a wireless video communication system that is a part of a complete wireless multimedia communication system. The International Telecommunication Union—Telecommunications Standardization Sector (ITU-T) H.223 [1] standard that describes a method of providing error protection to the video data before it is transmitted is also described. It should be noted that the main function of H.223 is to multiplex/demultiplex the audio, video, text, graphics, etc., which are typically communicated together in a videoconferencing application—error protection of the transmitted data becomes a requirement to support this functionality on error-prone channels. In Section 31.3, an overview of error-resilient video coding is given. The specific tools adopted into the International Standards Organization (ISO)/International Electrotechnical Commission (IEC) Motion Picture Experts Group (MPEG) v.4 (i.e., MPEG-4) [7] and the ITU-T H.263 [3] video coding standards to improve the error robustness of the video coder are described in Sections 31.4 and 31.5, respectively.

Table 31.1 provides a listing of some of the standards that are described or referred to in this chapter.

31.2 Wireless Video Communications

Figure 31.1 shows the basic block diagram of a wireless video communication system [10]. Input video is compressed by the video encoder to generate a compressed bitstream. The transport coder converts the compressed video bitstream into data units suitable for transmission over wireless channels. Typical operations carried out in the transport coder include channel coding, framing of data, modulation, and control operations required for accessing the wireless channel. At the receiver side, the inverse operations are performed to reconstruct the video signal for display.

In practice, the video communication system is part of a complete multimedia communication system and needs to interact with other system components to achieve the desired functionality. Hence, it becomes necessary to understand the other components of a multimedia communication system in order to design a good video communication system. Figure 31.2 shows the block diagram of a wireless multimedia terminal based on the ITU-T H.324 set of standards [4]. We use the H.324 standard as an example because it is the first videoconferencing standard for which mobile extensions were added to facilitate use on wireless channels. The system components of a multimedia terminal can be grouped into three processing blocks: (1) audio, video, and data (the word *data* is used here to mean still images/slides, shared files, documents etc.), (2) control, and (3) multiplex-demultiplex blocks.

TABLE 31.1 List of Relevant Standards

ISO/IEC 14496-2 (MPEG-4)	Information Technology—Coding of Audio-Visual Objects: Visual
H.263 (Version 1 and Version 2)	Video coding for low bitrate communication
H.261	Video codec for audiovisual services at p X 64 kbit/s
H.223	Multiplexing protocol for low bitrate multimedia communication
H.324	Terminal for low bitrate multimedia communication
H.245	Control protocol for multimedia communication
G.723.1	Dual rate speech coder for multimedia communication transmitting at 5.3 and 6.3 kbit/s

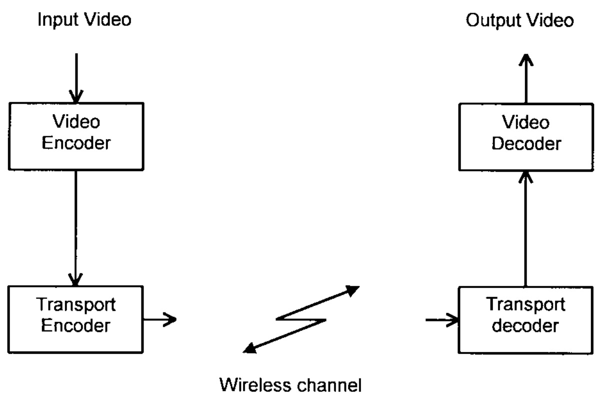

FIGURE 31.1: A wireless video communication system.

1. Audio, video, and data processing blocks—These blocks basically produce/consume the multimedia information that is communicated. The aggregate bitrate generated by these blocks is restricted due to limitations of the wireless channel and, therefore, the total rate allowed has to be judiciously allocated among these blocks. Typically, the video blocks use up the highest percentage of the aggregate rate, followed by audio and then data. H.324 specifies the use of H.261/H.263 for video coding and G.723.1 for audio coding.
2. Control block—This block has a wide variety of responsibilities all aimed at setting up and maintaining a multimedia call. The control block facilitates the set-up of compression methods and preferred bitrates for audio, video, and data to be used in the multimedia call. It is also responsible for end-to-network signalling for accessing the network and end-to-end signalling for reliable operation of the multimedia call. H.245 is the control protocol in the H.324 suite of standards that specifies the control messages to achieve the above functionality.
3. Multiplex-Demultiplex (MUX) block—This block multiplexes the resulting audio, video, data, and control signals into a single stream before transmission on the network. Similarly, the received bitstream is demultiplexed to obtain the audio, video, data, and control signals, which are then passed to their respective processing blocks. The MUX block accesses the network through a suitable network interface. The H.223 standard is the multiplexing scheme used in H.324.

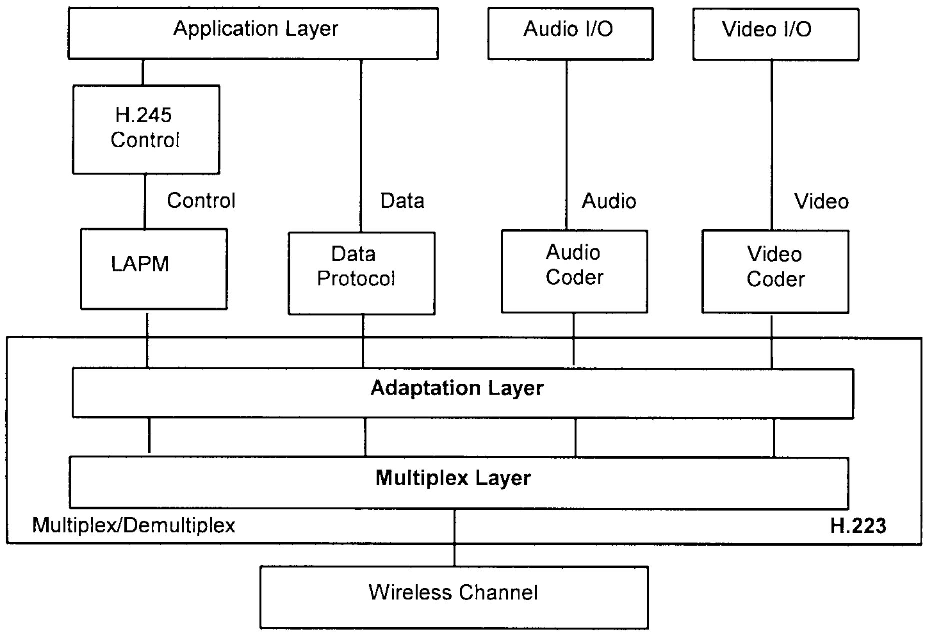

FIGURE 31.2: Configuration of a wireless multimedia terminal.

Proper functioning of the MUX is crucial to the operation of the video communication system, as all the multimedia data/signals flow through it. On wireless channels, transmission errors can lead to a breakdown of the MUX resulting in, for example, nonvideo data being channeled to the video decoder or corrupted video data being passed on to the video decoder. Three annexes were specifically added to H.223 to enable its operation in error-prone environments. Below, we give a more detailed overview of H.223 and point out the levels of error protection provided by H.223 and its three annexes. It should also be noted that MPEG-4 does not specify a lower-level MUX like H.223, and thus H.223 can also be used to transmit MPEG-4 video data.

31.2.1 Recommendation H.223

Video, audio, data, and control information is transmitted in H.324 on distinct logical channels. H.223 determines the way in which the logical channels are mixed into a single bitstream before transmission over the physical channel (e.g., the wireless channel). The H.223 multiplex consists of two layers—the multiplex layer and the adaptation layer, as shown in Fig. 31.2. The multiplex layer is responsible for multiplexing the various logical channels. It transmits the multiplexed stream in the form of packets. The adaptation layer adapts the information stream provided by the applications above it to the multiplex layer below it by adding, where appropriate, additional octets for the purposes of error control and sequence numbering. The type of error control used depends on the type of information (audio/video/data/control) being conveyed in the stream. The adaptation layer provides error control support in the form of both FEC and ARQ.

H.223 was initially targeted for use on the benign general switched telephone network (GSTN). Later on, to enable its use on wireless channels, three annexes (referred to as Levels 1–3, respectively), were defined to provide improved levels of error protection. The initial specification of H.223 is referred to as Level 0. Together, Levels 0–3 provide for a trade-off of error robustness against the overhead required, with Level 0 being the least robust and using the least amount of overhead and Level 3 being the most robust and also using the most amount of overhead.

1. H.223 Level 0—Default mode. In this mode the transmitted packet sizes are of variable length and are delimited by an 8-bit HDLC (High-level Data Link Control) flag (**01111110**). Each packet consists of a 1-octet header followed by the payload, which consists of a variable number of information octets. The header octet includes a Multiplex Code (MC) which specifies, by indexing to a multiplex table, the logical channels to which each octet in the information field belongs. To prevent emulation of the HDLC flag in the payload, bitstuffing is adopted.
2. H.223 Level 1 (Annex A)—Communication over low error-prone channels. The use of bitstuffing leads to poor performance in the presence of errors; therefore in Level 1, bitstuffing is not performed. The other improvement incorporated in Level 1 is the use of a longer 16-bit pseudo-noise synchronization flag to allow for more reliable detection of packet boundaries. The input bitstream is correlated with the synchronization flag and the output of the correlator is compared with a correlation threshold. Whenever the correlator output is equal to or greater than the threshold, a flag is detected. Since, bitstuffing is not performed, it is possible to have this flag emulated in the payload. However, the probability of such an emulation is low and is outweighed by the improvement gained by not using bitstuffing over error-prone channels.
3. H.223 Level 2 (Annex B)—Communication over moderately error-prone channels. When compared to the Level 1 operation, Level 2 increases the protection on the packet header. A Multiplex Payload Length (MPL) field, which gives the length of the payload in bytes, is introduced into the header to provide additional redundancy for detecting the length of the video packet. A (24,12,8) extended Golay code is used to protect the MC and the MPL fields. Use of error protection in the header enables robust delineation of packet boundaries. Note that the payload data is not protected in Level 2.
4. H.223 Level 3 (Annex C)—Communication over highly error-prone channels. Level 3 goes one step above Level 2 and provides for protection of the payload data. Rate Compatible Punctured Convolutional (RCPC) codes, various CRC polynomials, and ARQ techniques are used for protection of the payload data. Level 3 allows for the payload error protection overhead to vary depending on the channel conditions. RCPC codes are used for achieving this adaptive level of error protection because RCPC codes use the same channel decoder architecture for all the allowed levels of error protection, thereby reducing the complexity of the MUX.

31.3 Error Resilient Video Coding

Even after error control and correction, some amount of residual errors still exist in the compressed bitstream fed to the video decoder in the receiver. Therefore, the video decoder should be robust to these errors and should provide acceptable video quality even in the presence of some residual errors. In this section, we first describe a standard video coder configuration that is the basis of many international standards and also highlight the potential problems that are encountered when compressed video from these systems is transmitted over wireless channels. We then give an overview of the strategies that can be adopted to overcome these problems. Most of these strategies are incorporated in the MPEG-4 video coding standard and the H.263 (Version 2) video coding standard [3]. The original H.263 standard [2] which was standardized in 1996 for use in H.324 terminals connected to GSTN is referred to as Version 1. Version 2 of the H.263 standard provides additional improvements and functionalities (which include error-resilience tools) over the Version 1 standard. We will use H.263 to refer to both Version 1 and Version 2 standards and a distinction will be made only when required.

31.3.1 A Standard Video Coder

Redundancy exists in video signals in both spatial and temporal dimensions. Video coding techniques exploit this redundancy to achieve compression. A plethora of video compression techniques have been proposed in the literature, but a hybrid coding technique consisting of block motion compensation (BMC) and discrete cosine transforms (DCT) has been found to be very effective in practice. In fact, most of the current video coding standards such as H.263 and MPEG-4, which provide state-of-the-art compression performance, are all based on this hybrid coding technique. In this hybrid BMC/DCT coding technique, BMC is used to exploit temporal redundancy and the DCT is used to reduce spatial redundancy.

Figure 31.3 illustrates a standard hybrid BMC/DCT video coder configuration. Pictures are coded in either of two modes—interframe (INTER) or intraframe (INTRA) mode. In intraframe coding, the video image is encoded without any relation to the previous image, whereas in interframe coding, the current image is predicted from the previous image using BMC, and the difference between the current image and the predicted image, called the residual image, is encoded. The basic unit of data which is operated on is called a macroblock (MB) and is the data (both **luminance and chrominance** components) corresponding to a block of 16×16 pixels. The input image is split into disjoint macroblocks and the processing is done on a macroblock basis. Motion information, in the form of **motion vectors,** is calculated for each macroblock. The motion compensated prediction residual error is then obtained by subtracting each pixel in the macroblock with its motion shifted counterpart in the previous frame. Depending on the mode of coding used for the macroblock, either the image macroblock or the corresponding residual image macroblock is split into blocks of size 8×8 and an 8×8 DCT is applied to each of these 8×8 blocks. The resulting DCT coefficients are then quantized. Depending on the quantization step-size, this will result in a significant number of zero-valued coefficients. To efficiently encode the DCT coefficients that remain nonzero after quantization, the DCT coefficients are zig-zag scanned, and run-length encoded and the run-lengths are variable length encoded before transmission. Since a significant amount of correlation exists between the neighboring macroblocks' motion vectors, the motion vectors are themselves predicted from already transmitted motion vectors and the motion vector prediction error is encoded. The motion vector prediction error and the mode information are also variable length coded before transmission to achieve efficient compression.

The decoder uses a reverse process to reconstruct the macroblock at the receiver. The variable length codewords present in the received video bitstream are decoded first. For INTER macroblocks, the pixel values of the prediction error are reconstructed by inverse quantization and inverse DCT and are then added to the motion compensated pixels from the previous frame to reconstruct the transmitted macroblock. For INTRA macroblocks, inverse quantization and inverse DCT directly result in the transmitted macroblock. All macroblocks of a given picture are decoded to reconstruct the whole picture.

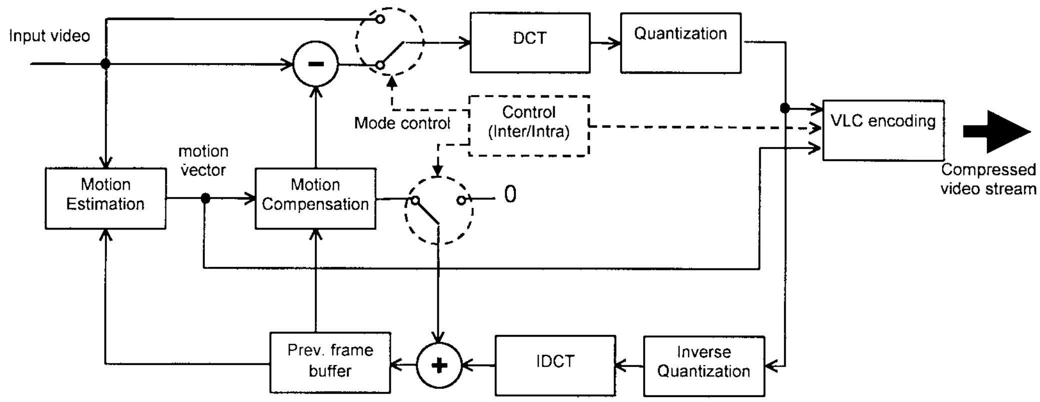

FIGURE 31.3: A standard video coder.

31.3.2 Error Resilient Video Decoding

The use of predictive coding and variable length coding (VLC), though very effective from a compression point of view, makes the video decoding process susceptible to transmission errors. In VLC, the boundary between codewords is implicit. The compressed bitstream has to be read until a full codeword is encountered; the codeword is then decoded to obtain the information encoded in the codeword. When there are transmission errors, the implicit nature of the boundary between codewords typically leads to an incorrect number of bits being used in VLC decoding and, thus, subsequently results in a loss of synchronization with the encoder. In addition, the use of predictive coding leads to the propagation of these transmission errors to neighboring spatial blocks and to subsequently decoded frames, which leads to a rapid degradation in the reconstructed video quality.

To minimize the disastrous impact that transmission errors can have on the video decoding process, the following stages are incorporated in the video decoder to make it more robust:

- Error detection and localization
- Resynchronization
- Data recovery
- Error concealment

Figure 31.4 shows an error resilient video decoder configuration. The first step involved in robust video coding is the detection of errors in the bitstream. The presence of errors in the bitstream can be signaled by the FEC used in the multiplex layer. The video coder can also detect errors whenever illegal VLC codewords are encountered in the bitstream or when the decoding of VLC codewords leads to an illegal value of the decoded information (e.g., occurrence of more than 64 DCT coefficients for an 8×8 DCT block). Accurate detection of errors in the bitstream is a very important step, since most of the other error resilience techniques can only be invoked if an error is detected.

Due to the use of VLC, the location in the bitstream where the decoder detects an error is not the same location where the error has actually occurred but some undetermined distance away from it. This is shown in Fig. 31.5. Once an error is detected, it also implies that the decoder is not in synchronization with the encoder. Resynchronization schemes are then employed for the decoder to fall back into lock step with the encoder. While constructing the bitstream, the encoder inserts unique resynchronization words into the bitstream at approximately equally spaced intervals. These

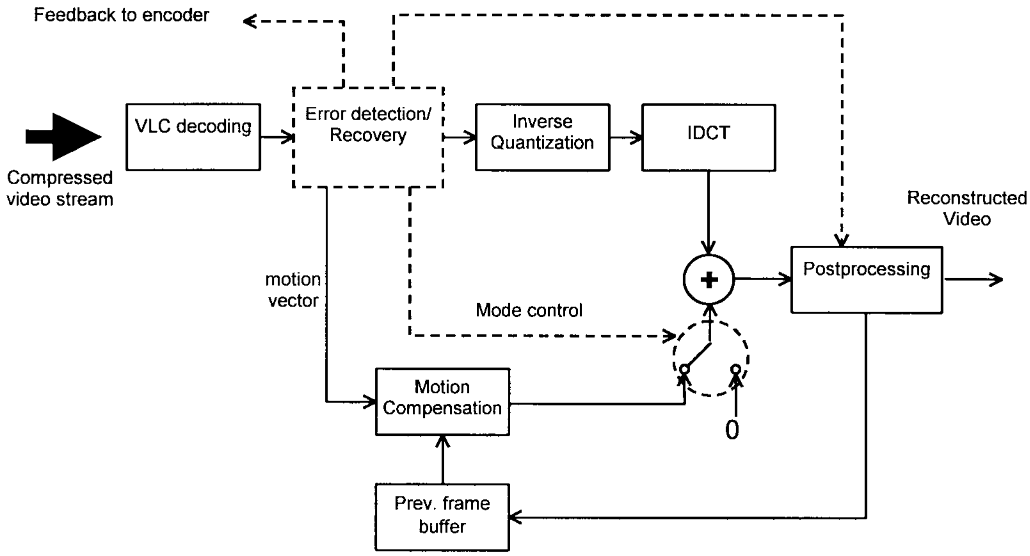

FIGURE 31.4: Error resilient video decoder.

resynchronization words are chosen such that they are unique from the valid video bitstream. That is, no valid combination of the video algorithm's VLC tables can produce these words. The decoder, upon detection of an error, seeks forward in the bitstream looking for this known resynchronization word. Once this word is found, the decoder then falls back in synchronization with the encoder. At this point, the decoder has detected an error, regained synchronization with the encoder, and isolated the error to be between the two resynchronization points. Since the decoder can only isolate the error to be somewhere between the resynchronization points but not pinpoint its exact location, all of the data that corresponds to the macroblocks between these two resynchronization points needs to be discarded. Otherwise, the effects of displaying an image reconstructed from erroneous data can cause highly annoying visual artifacts.

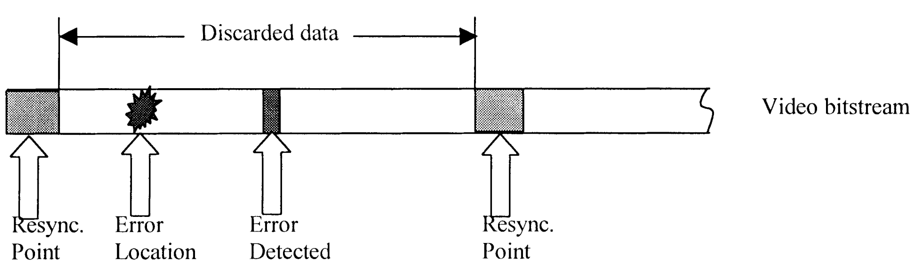

FIGURE 31.5: At the decoder, it is usually not possible to detect the error at the actual error occurrence location; hence, all the data between the two resynchronization points may need to be discarded.

Some data recovery techniques, such as "reversible decoding," enable the decoder to salvage some of the data between the two resynchronization points. These techniques advocate the use of a special kind of VLC table at the encoder in coding the DCTs and motion vector information. These special VLCs have the property that they can be decoded both in the forward and reverse directions. By comparing the forward and reverse decoded data, the exact location of the error in the bit stream can be localized more precisely and some of the data between the two resynchronization points can be salvaged. The use of these reversible VLCs (RVLCs) is part of the MPEG-4 standard and will be described in greater detail in the following sections.

After data recovery, the impact of the data that is deemed to be in error needs to be minimized. This is the error concealment stage. One simple error concealment strategy is to simply replace the luminance and chrominance components of the erroneous macroblocks with the luminance and chrominance of the corresponding macroblocks in the previous frame of the video sequence. While this technique works fairly well and is simple to implement, more complex techniques use some type of estimation strategies to exploit the local correlation that exists within a frame of video data to come up with a better estimate of the missing or erroneous data. These error concealment strategies are essentially postprocessing algorithms and are not mandated by the video coding standards. Different implementations of the wireless video systems utilize different kinds of error concealment strategies based on the available computational power and the quality of the channel.

If there is support for a decoder feedback path to the encoder as shown in Fig. 31.3, this path can be used to signal detected errors. The feedback information from the decoder can be used to retransmit data or to influence future encoder action so as to stop the propagation of detected errors in the decoder. Note that for the feedback to take place, the network must support a back channel.

31.3.3 Classification of Error-Resilience Techniques

In general, techniques to improve the robustness of the video coder can be classified into three categories based on whether the encoder or the decoder plays a primary part in improving the error robustness [10]. *Forward error resilience* techniques refer to those techniques where the encoder plays the primary part in improving the error robustness, typically by introducing redundancy in the transmitted information. In *postprocessing* techniques, the decoder plays the primary part and does concealment of errors by estimation and interpolation (e.g., spatial-temporal filtering) using information it has already received. In *interactive error* resilience techniques, the decoder and the encoder interact to improve the error resilience of the video coder. Techniques that use decoder feedback come under this category.

31.4 MPEG-4 Error Resilience Tools

MPEG-4 is an ISO/IEC standard being developed by the Motion Pictures Expert Group. Initially MPEG was aimed primarily at low-bit-rate communications; however, its scope was later expanded to be much more of a multimedia coding standard [7]. The MPEG-4 video coding standard is the first video coding standard to address the problem of efficient representation of visual objects of arbitrary shape. MPEG-4 was also designed to provide "universal accessibility," i.e., the ability to access audio-visual information over a wide range of storage and transmission media. In particular, because of the proliferation of wireless communications, this implied development of specific tools to enable error-resilient transmission of compressed data over noisy communication channels.

A number of tools have been incorporated into the MPEG-4 video coder to make it more error resilient. All these tools are basically forward error resilience tools. We describe below each of these tools and its advantages.

31.4.1 Resynchronization

As mentioned earlier, a video decoder that is decoding a corrupted bitstream may lose synchronization with the encoder (i.e., it is unable to identify the precise location in the image where the current data belongs). If remedial measures are not taken, the quality of the decoded video rapidly degrades and becomes unusable. One approach is for the encoder to introduce resynchronization markers in the bitstream at various locations. When the decoder detects an error, it can then look for this resynchronization marker and regain synchronization.

Previous video coding standards such as H.261 and H.263 (Version 1) logically partition each of the images to be encoded into rows of macroblocks called Group Of Blocks (GOBs). These GOBs correspond to a horizontal row of macroblocks for **QCIF** images. Figure 31.6 shows the GOB numbering scheme for H.263 (Version 1) for QCIF resolution. For error resilience purposes, H.263 (Version 1) provides the encoder an option of inserting resynchronization markers at the beginning of each of the GOBs. Hence, for QCIF images these resynchronization markers are allowed to occur only at the left edge of the images. The smallest region that the error can be isolated to and concealed in this case is thus one row of macroblocks.

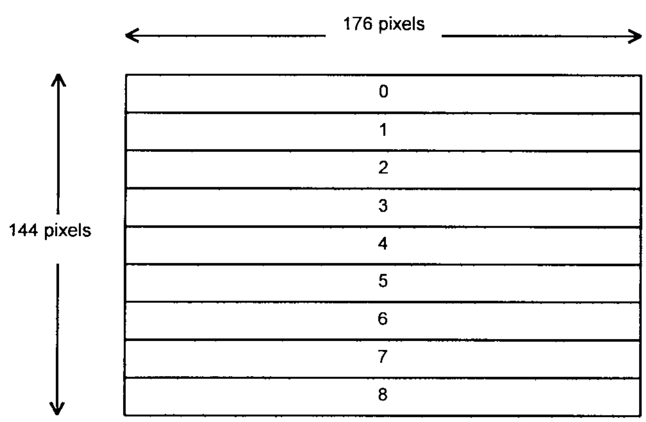

FIGURE 31.6: H.263 GOB numbering for a QCIF image.

In contrast, the MPEG-4 encoder is not restricted to inserting the resynchronization markers only at the beginning of each row of macroblocks. The encoder has the option of dividing the image into video packets. Each video packet is made up of an integer number of consecutive macroblocks in raster scan order. These macroblocks can span several rows of macroblocks in the image and can even include partial rows of macroblocks. One suggested mode of operation for the MPEG-4 encoder is for it to insert a resynchronization marker periodically at approximately every K bits. Note that resynchronization markers can only be placed at a macroblock boundary and, hence, the video packet length cannot be constrained to be exactly equal to K bits. When there is a significant activity in one part of the image, the macroblocks corresponding to these areas generate more bits than other parts of the image. If the MPEG-4 encoder inserts the resynchronization markers at uniformly spaced bit intervals, the macroblock interval between the resynchronization markers is a lot closer in the high activity areas and a lot farther apart in the low activity areas. Thus, in the presence of a short burst of errors, the decoder can quickly localize the error to within a few macroblocks in the important high activity areas of the image and preserve the image quality in these important areas. In the case of H.263 (Version 1), where the resynchronization markers are restricted to be at the beginning of the GOBs, it is only possible for the decoder to isolate the errors to a fixed GOB independent of the image content. Hence, effective coverage of the resynchronization marker is reduced when compared to the MPEG-4 scheme. The recommended spacing of the resynchronization markers in MPEG-4 is based on the bitrates. For 24 Kb/s, it is recommended to insert them at intervals of 480 bits and for bitrates between 25 Kb/s to 48 Kb/s, it is recommended to place them at every 736 bits. Figures 31.7(a) and (b) illustrate the placement of resynchronization markers for H.263 (Version 1) and MPEG-4.

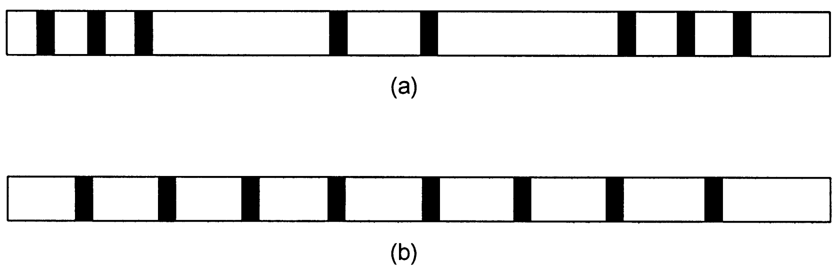

FIGURE 31.7: Position of resynchronization markers in the bitstream for (a) H.263 (Version 1) encoder with GOB headers and for (b) an MPEG-4 encoder with video packets.

Note that in addition to inserting the resynchronization markers at the beginning of each video packet, the encoder also needs to remove all data dependencies that exist between the data belonging to two different video packets within the same image. This is required so that even if one of the video packets in the current image is corrupted due to errors, the other packets can be decoded and utilized by the decoder. In order to remove these data dependencies, the encoder inserts two additional fields in addition to the resynchronization marker at the beginning of each video packet, as shown in Fig. 31.8. These are, (1) the absolute macroblock number of the first macroblock in the video packet, *Mb. No.*, (which indicates the spatial location of the macroblock in the current image), (2) the quantization parameter, QP, which denotes the initial quantization parameter used to quantize the DCT coefficients in the video packet. The encoder also modifies the predictive encoding method used for coding the motion vectors such that there are no predictions across the video packet boundaries. Also shown in Fig. 31.8 is a third field, labeled *HEC*. Its use is discussed in a later section.

Resync. Marker	MB No.	QP	HEC	Macroblock data

FIGURE 31.8: An MPEG-4 video packet.

31.4.2 Data Partitioning

Data partitioning in MPEG-4 provides enhanced error localization and error concealment capabilities. The data partitioning mode partitions the data within a video packet into a motion part and a texture part (DCT coefficients) separated by a unique Motion Marker (MM), as shown in Fig. 31.9. All the syntactic elements of the video packet that have motion-related information are placed in the motion partition and all the remaining syntactic elements that relate to the DCT data are placed in the texture partition. If the texture information is lost, data partitioning enables the salvation of motion information, which can then be used to conceal the errors in a more effective manner.

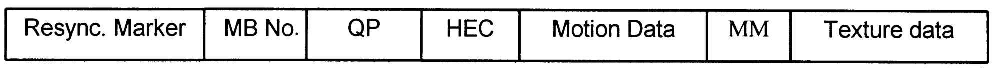

FIGURE 31.9: A data partitioned MPEG-4 video packet.

The motion marker is computed from the motion VLC tables using a search program such that it is Hamming distance 1 from any possible valid combination of the motion VLC tables [9]. The motion marker is uniquely decodable from the motion VLC tables, and it indicates to the decoder the end of the motion information and the beginning of the DCT information. The number of macroblocks in the video packet is implicitly known after encountering the motion marker. Note that the motion marker is only computed once based on the VLC tables and is fixed in the standard. Based on the VLC tables in MEPG-4, the motion marker is a 17-bit word whose value is **1 1111 0000 0000 0001.**

31.4.3 Reversible Variable Length Codes (RVLCs)

As was shown in Fig. 31.5, if the decoder detects an error during the decoding of VLC codewords, it loses synchronization and hence typically has to discard all the data up to the next resynchronization point. RVLCs are designed such that they can be instantaneously decoded both in the forward and the backward direction. When the decoder detects an error while decoding the bitstream in the forward direction, it jumps to the next resynchronization marker and decodes the bitstream in the backward direction until it encounters an error. Based on the two error locations, the decoder can recover some of the data that would have otherwise been discarded. This is shown in Fig. 31.10, which shows only the texture part of the video packet—only data in the shaded area is discarded. Note that if RVLCs were not used, all the data in the texture part of the video packet would have to be discarded. RVLCs thus enable the decoder to better isolate the error location in the bitstream.

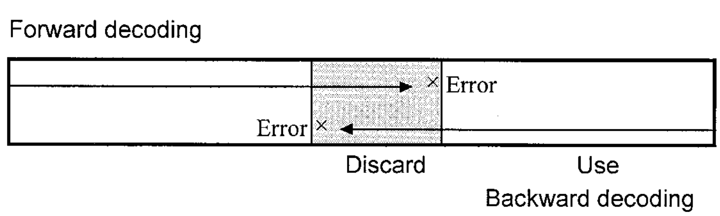

FIGURE 31.10: Use of reversible variable length codes.

Figure 31.11 shows the comparison of performance of resynchronization, data partitioning, and RVLC techniques for 24 Kb/s QCIF video data. The experiments involved transmission of three video sequences, each of duration 10s, over a bursty channel simulated by a 2-state Gilbert model [6]. The burst duration on the channel is 1 ms and the burst occurrence probability is 10^{-2}. Figure 31.11, which plots the average peak signal-to-noise ratios of the received video frames, shows that data partitioning and RVLC provide improved performance when compared to using only resynchronization markers.

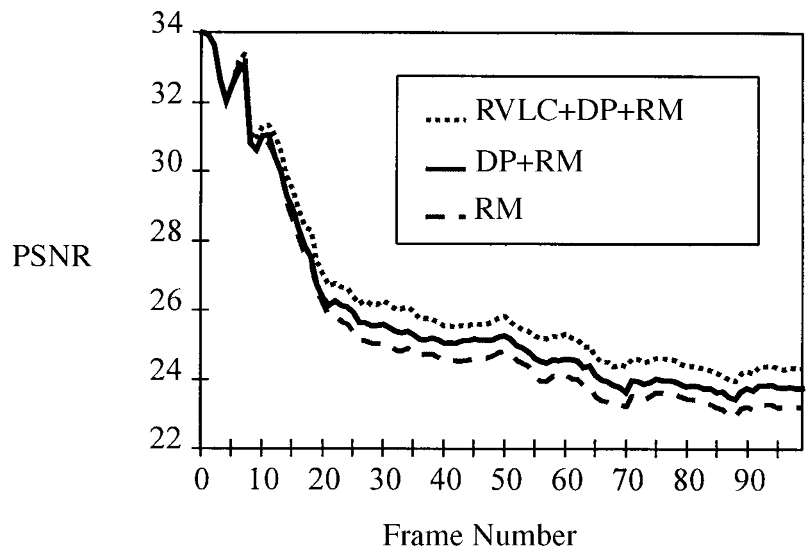

FIGURE 31.11: Performance comparison of resynchronization, data partitioning, and RVLC over a bursty channel simulated by a 2-state Gilbert model. Burst durations are 1ms long and the probability of occurrence of a burst is 10^{-2}. Legend: RM—resynchronization marker; DP—data partitioning; RVLC—reversible variable length codes.

31.4.4 Header Extension Code (HEC)

Some of the most important information that the decoder needs in order to decode the video bitstream is in the video frame header data. This data includes information about the spatial dimensions of the video data, the time stamps associated with the decoding and the presentation of this video data, and the type of the current frame (INTER/INTRA). If some of this information is corrupted due to channel errors, the decoder has no other recourse but to discard all the information belonging to the current video frame. In order to reduce the sensitivity of this data, a technique called Header Extension Code (HEC) was introduced into the MPEG-4 standard. In each video packet, a 1-bit field called HEC is present. The location of HEC in the video packet is shown in Fig. 31.8. For each video packet, when HEC is set, the important header information that describes the video frame is repeated in the bits following the HEC. This information can be used to verify and correct the header information of the video frame. The use of HEC significantly reduces the number of discarded video frames and helps achieve a higher overall decoded video quality.

31.4.5 Adaptive Intra Refresh (AIR)

Whenever an INTRA macroblock is received, it basically stops the temporal propagation of errors at its corresponding spatial location. The procedure of forcefully encoding some macroblocks in a frame in INTRA mode to flush out possible errors is called INTRA refreshing. INTRA refresh is very effective in stopping the propagation of errors, but it comes at the cost of a large overhead. Coding a macroblock in INTRA mode typically requires many more bits when compared to coding the macroblock in INTER mode. Hence, the INTRA refresh technique has to be used judiciously.

For areas with low motion, simple error concealment by just copying the previous frame's macroblocks works quite effectively. For macroblocks with high motion, error concealment becomes very difficult. Since the high motion areas are perceptually the most significant, any persistent error in the high motion area becomes very noticeable. The AIR technique of MPEG-4 makes use of the above facts and INTRA refreshes the motion areas more frequently, thereby allowing the corrupted high motion areas to recover quickly from errors.

Depending on the bitrate, the AIR approach only encodes a fixed and predetermined number of macroblocks in a frame in INTRA mode (the exact number is not standardized by MPEG-4). This fixed number might not be enough to cover all the macroblocks in the motion area; hence, the AIR technique keeps track of the macroblocks that have been refreshed (using a "refresh map") and in subsequent frames refreshes any macroblocks in the motion areas that might have been left out.

31.5 H.263 Error Resilience Tools

In this section, we discuss four error resilience techniques which are part of the H.263 standard—*slice structure mode* and *independent segment decoding*, which are forward error resilience features, and *error tracking* and *reference picture selection*, which are interactive error resilience techniques. Error tracking was introduced in H.263 (Version 1) as an appendix, whereas the remaining three techniques were introduced in H.263 (Version 2) as annexes.

31.5.1 Slice Structure Mode (Annex K)

The slice structured mode of H.263 is similar to the video packet approach of MPEG-4 with a slice denoting a video packet. The basic functionality of a slice is the same as that of a video packet—providing periodic resynchronization points throughout the bistream. The structure of a slice is shown in Fig. 31.12. Like an MPEG-4 video packet, the slice consists of a header followed by the macroblock data. The SSC is the slice start code and is identical to the resynchronization marker of MPEG-4. The MBA field, which denotes the starting macroblock number in the slice, and the SQUANT field, which is the quantizer scale coded nonpredictively, allow for the slice to be coded independently.

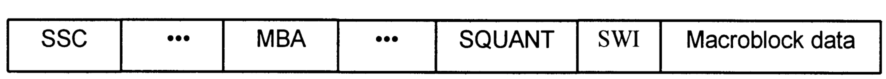

FIGURE 31.12: Structure of a slice in H.263/Annex K.

The slice structured mode also contains two submodes which can be used to provide additional functionality. The submodes are

- Rectangular slice submode (RSS)—This allows for rectangular shaped slices. The rectangular region contained in the slice is specified by SWI+1 (See Fig. 31.12 for the location of the SWI field in the slice header), which gives the width of the rectangular region, and MBA, which specifies the upper left macroblock of the slice. Note that the height of the rectangular region gets specified by the number of macroblocks contained in the slice. This mode can be used, for example, to subdivide images into rectangular regions of interest for region-based coding.
- Arbitrary slice submode (ASO)—The default order of transmission of slices is such that the MBA field is strictly increasing from one slice to the next transmitted slice. When ASO is used, the slices may appear in any order within the bitstream. This mode is useful when the wireless network supports prioritization of slices which might result in out-of-order arrival of video slices at the decoder.

31.5.2 Independent Segment Decoding (ISD) (Annex R)

Even though the slice structured mode eliminates decoding dependency between neighboring slices, errors in slices can spatially propagate to neighboring slices in subsequent frames due to motion compensation. This happens because motion vectors in a slice can point to macroblocks of neighboring slices in the reference picture. Independent segment decoding eliminates this from happening by restricting the motion vectors within a predefined segment of the picture from pointing to other segments in the picture, thereby helping to contain the error to be within the erroneous segment. This improvement in the localization of errors, however, comes at a cost of a loss of coding efficiency. Because of this restriction on the motion vectors, the motion compensation is not as effective, and the residual error images use more bits.

For ease of implementation, the ISD mode puts restrictions on segment shapes and on the changes of segment shapes from picture to picture. The ISD mode cannot be used with the slice structured mode (Annex K) unless the rectangular slice submode of Annex K is active. This prevents the need for treating awkward shapes of slices that can otherwise arise when Annex K is not used with rectangular slice submode. The segment shapes are not allowed to change from picture to picture unless an INTRA frame is being coded.

31.5.3 Error Tracking (Appendix I)

The error tracking approach is an INTRA refresh technique but uses decoder feedback of errors to decide which macroblocks in the current image to code in INTRA mode to prevent the propagation of these errors. When there are no errors on the channel, normal coding (which usually results in the bit-efficient INTER mode being selected most of the time) is used. The use of decoder feedback allows the system to adapt to varying channel conditions and minimizes the use of forced INTRA updates to situations when there are channel errors.

Because of the time delay involved in the decoder feedback, the encoder has to track the propagation of an error from its original occurrence to the current frame to decide which macroblocks should be INTRA coded in the current frame. A low complexity algorithm was proposed in Appendix I of H.263 to track the propagation of errors. However, it should be noted that the use of this technique is not mandated by H.263. Also, H.263 itself does not standardize the mechanism by which the decoder feedback of error can be sent. Typically, H.245 control messages are used to signal the decoder feedback for error tracking purposes.

31.5.4 Reference Picture Selection (Annex N)

The Reference Picture Selection (RPS) mode of H.263 also relies on decoder feedback to efficiently stop the propagation of errors. The back channel used in RPS mode can be a separate logical channel (e.g., by using H.245), or if two-way communication is taking place, the back channel messages can be sent multiplexed with the encoded video data. In the presence of errors, the RPS mode allows the encoder to be instructed to select one of the several previously correctly received and decoded frames as the reference picture for motion compensation of the current frame being encoded. This effectively stops the propagation of error. Note that the use of RPS requires the use of multiple frame buffers at both the encoder and the decoder to store previously decoded frames. Hence, the improvement in performance in the RPS mode has come at the cost of increased memory requirements.

31.6 Discussion

In this chapter we presented a broad overview of the various techniques that enable wireless video transmission. Due to the enormous amount of bandwidth required, video data is typically compressed before being transmitted, but the errors introduced by the wireless channels have a severe impact on the compressed video information. Hence, special techniques need to be employed to enable robust video transmission. International standards play a very important role in communications applications. The two current standards that are most relevant to video applications are ISO MPEG-4 and ITU H.263. In this chapter, we detailed these two standards and explained the error resilient tools that are part of these standards to enable robust video communication over wireless channels. A tutorial overview of these tools has been presented and the performance of these tools has been described.

There are, however, a number of other methods that further improve the performance of a wireless video codec that the standards do not specify. If the encoder and decoder are aware of the limitations imposed by the communication channel, they can further improve the video quality by using these methods. These methods include encoding techniques such as rate control to optimize the allocation of the effective channel bit rate between various parts of video to be transmitted and intelligent decisions on when and where to place INTRA refresh macroblocks to limit the error propagation. Decoding methods such as superior error concealment strategies that further conceal the effects of erroneous macroblocks by estimating them from correctly decoded macroblocks in the spatiotemporal neighborhood can also significantly improve the effective video quality.

This chapter has mainly focused on the error resilience aspects of the video layer. There are a number of error detection and correction strategies, such as Forward Error Correction (FEC), that can further improve the reliability of the transmitted video data. These FEC codes are typically provided in the systems layer and the underlying network layer. If the video transmission system has the ability to monitor the dynamic error characteristics of the communication channel, joint source-channel coding techniques can also be effectively employed. These techniques enable the wireless communication system to perform optimal trade-offs in allocating the available bits between the source coder (video) and the channel coder (FEC) to achieve superior performance.

Current video compression standards also support *layered* coding methods. In this approach, the compressed video information can be separated into multiple layers. The *base* layer, when decoded, provides a certain degree of video quality and the *enhancement* layer, when received and decoded, then adds to the base layer to further improve the video quality. In wireless channels, these base and enhancement layers give a natural method of partitioning the video data into more important and less important layers. The base layer can be protected by a stronger level of error protection (higher overhead channel coder) and the enhancement layer by a lesser strength coder. Using this

Unequal Error Protection (UEP) scheme, the communication system is assured of a certain degree of performance most of the time through the base layer, and when the channel is not as error prone and the decoder receives the enhancement layer, this scheme provides improved quality.

Given all these advances in video coding technology, coupled with the technological advances in processor technology, memory devices, and communication systems, wireless video communications is fast becoming a very compelling application. With the advent of higher bandwidth third generation wireless communication systems, it will be possible to transmit compressed video in many wireless applications, including mobile videophones, videoconferencing systems, PDAs, security and surveillance applications, mobile Internet terminals, and other multimedia devices.

Defining Terms

Automatic Repeat reQuest (ARQ): An error control system in which notification of erroneously received messages is sent to the transmitter which then simply retransmits the message. The use of ARQ requires a feedback channel and the receiver must perform error detection on received messages. Redundancy is added to the message before transmission to enable error detection at the receiver.

Block motion compensation (BMC): Motion compensated prediction that is done on a block basis; that is, blocks of pixels are assumed to be displaced spatially in a uniform manner from one frame to another.

Forward Error Correction: Introduction of redundancy in data to allow for correction of errors without retransmission.

Luminance and chrominance: Luminance is the brightness information in a video image, whereas chrominance is the corresponding color information.

Motion vectors: Specifies the spatial displacement of a block of pixels from one frame to another.

QCIF: Quarter Common Intermediate Format (QCIF) is a standard picture format that defines the image dimensions to be 176×144 (pixels per line $\times$ lines per picture) for luminance and 88×72 for chrominance.

References

[1] International Telecommunications Union—Telecommunications Standardization Sector, Recommendation H.223: Multiplexing protocol for low bitrate multimedia communications, Geneva, 1996.

[2] International Telecommunications Union—Telecommunications Standardization Sector, Recommendation H.263: Video coding for low bitrate communication, Geneva, 1996.

[3] International Telecommunications Union—Telecommunications Standardization Sector, Draft Recommendation H.263 (Version 2): video coding for low bitrate communication, Geneva, 1998.

[4] International Telecommunications Union—Telecommunications Standardization Sector, Recommendation H.324: terminal for low bit rate multimedia communications, Geneva, 1996.

[5] Lin, S. and Costello, D.J., Jr., *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[6] Miki, T., et al., Revised error pattern generation programs for core experiments on error resilience, *ISO/IEC JTC1/SC29/WG11 MPEG96/1492,* Maceio, Brazil, Nov. 1996.

[7] International Organization for Standardization, Committee draft of Tokyo (N2202): information technology—coding of audio-visual objects: visual, *ISO/IEC 14496-2*, Mar. 1998.

[8] Sklar, B., Rayleigh fading channels in mobile digital communication systems, Pt. I: Characterization, *IEEE Commun. Mag.*, 35, 90–100, 1997.

[9] Talluri, R., et al., Error concealment by data partitioning, to appear in *Signal Processing: Image Commun.*, 1998.

[10] Wang, Y. and Zhu, Q., Error control and concealment for video communication: a review, *IEEE Trans. Circuits Syst. Video Technol.*, 86(5), 974–997, May 1998.

Further Information

A broader overview of wireless video can be found in the special issue of *IEEE Communications Magazine*, June 1998. Wang and Zhu [10] provide an exhaustive review of error concealment techniques for video communications. More details on MPEG-4 and ongoing Version 2 activities in MPEG-4 can be found on the web page `http://drogo.cselt.it/mpeg/standards/mpeg-4/mpeg-4.htm`. H.263 (Version 2) activities are tracked on the web page `http://www.ece.ubc.ca/spmg/research/motion/h263plus/`. Most of the ITU-T recommendations can be obtained from the web site `http://www.itu.org`. The special issue of *IEEE Communications Magazine*, December 1996, includes articles on H.324 and H.263.

Current research relevant to wireless video communications is reported in a number of journals including *IEEE Transactions on Circuits and Systems for Video Technology, IEEE Transactions on Image Processing, IEEE Transactions on Vehicular Technology, Signal Processing: Image Communication*. The *IEEE Communications Magazine* regularly reports review articles relevant to wireless video communications. Conferences of interest include the IEEE International Conference on Image Processing (ICIP), IEEE Vehicular Technology Conference (VTC), and IEEE International Conference on Communications (ICC).