Robust Video Coding Algorithms and Systems

JOHN D. VILLASENOR, SENIOR MEMBER, IEEE,
YA-QIN ZHANG, FELLOW, IEEE, AND JIANGTAO WEN, ASSOCIATE MEMBER, IEEE

Invited Paper

Wireless video communication is particularly challenging because it combines the already difficult problem of efficient compression with the additional and usually contradictory need to make the compressed bit stream robust to channel errors. We describe design and implementation strategies for error-robust video communications with an emphasis on techniques compatible with the coding approaches used in the ISO (MPEG-4) and ITU standards organizations. These techniques include modifications to the video coding algorithms as well as to the system layers that perform packetization and multiplexing.

Keywords— *Error concealment, MPEG-4, robust communications, video compression.*

I. INTRODUCTION

Delivery of real-time video in the presence of constraints on bandwidth, delay, complexity, and channel reliability is one of the most interesting and relevant contemporary communications problems. While the above constraints are present in many communications systems, the challenges they impose are particularly acute for real-time video. In contrast with speech, which can be coded using fixed rate algorithms operating in the 5–12-kbit/s range, “low-bit rate” video coding involves rates ranging from tens to hundreds of kilobits per second or more and is inherently a variable-rate process. In contrast with data, which are not usually subject to strict delay constraints and can therefore be handled using network protocols that use retransmission to ensure error-free delivery, real-time video is delay sensitive by definition and cannot easily make use of retransmission. The variable-rate nature of video and the extensive use of entropy coding in video coding renders compressed video especially vulnerable to errors, and successful video communication in the presence of errors requires careful design strategies at the encoder, decoder, and other system layers.

In view of the above, it is not surprising that while there is a large and growing commercial infrastructure for the delivery of wireless voice and data, wireless video is still largely absent from the commercial scene. However, it is becoming increasingly clear that wireless video will play an important role in emerging and future generations of communications systems. One impetus for this will be the growing access to high-quality voice, data, and video information via wireline systems. As has already occurred with voice, this will create market pressure to offer similar services in wireless environments. Another factor will be the growing availability of low-power digital signal processors (DSP’s) and microprocessors capable of performing video compression in handheld, battery-powered terminals. It is only in the last few years that it has become practical to do real-time video coding at acceptable quality in less power-constrained environments such as commercial microprocessors used in PC’s. The near future will see the development of DSP’s operating with supply voltages of about 1 V which will for the first time make it practical to perform video coding in portable wireless systems. Yet another impetus for wireless video will come from the wireless service providers. While today it is rare to have access to wide-area wireless data services at rates higher than 10 kbit/s, organizations including the Telecommunications Institute of America, ETSI, and the International Telecommunications Union (ITU) through the IMT-2000 effort are developing real-time high-speed wireless data services that are likely to offer approximately 60 kbit/s within the next two or three years, and substantially higher bandwidths in subsequent years. When these services are deployed, there will inevitably be interest in using them for real-time visual communications. In addition, the last several years have seen important convergence of low bit rate video coding standardization efforts, most notably between ITU/H.324 and International Standards Organization (ISO) MPEG-4. In addition to meeting the basic goal of enabling interoperability, these standards have an array of features that can be used to support robust transmission of low-rate video in error-prone environments. Finally, there

Manuscript received December 19, 1997; revised December 17, 1998.

J. D. Villasenor is with the Electrical Engineering Department, University of California, Los Angeles, CA 90024-1594 USA.

Y.-Q. Zhang is with Microsoft Research China, Beijing Sigma Center, Beijing 100080 China.

J. Wen is with PacketVideo Technology, San Diego, CA 92121 USA.

Publisher Item Identifier S 0018-9219(99)07546-5.

are many scenarios where the availability of efficient, error-robust wireless video would constitute a useful and in some instances lifesaving resource. These applications including emergency medicine, security monitoring, military use, as well as "traditional" videoconferencing.

The remainder of this paper is organized as follows. In Section II we present a brief overview of the ITU and ISO video coding standards and identify features of the algorithms which are particularly relevant to error robustness. In Section III we discuss algorithmic modifications that lead to improved video robustness at little or no cost to coding efficiency. Section IV describes robust multiplexing, drawing heavily on the recently completed work of the ITU to develop more robust versions of the H.324 low bit-rate multimedia terminal. Section V describes experimental wireless video testbeds with emphasis on the Handheld Multimedia Terminals (HMT) and Wireless Internetworking Testbed (WIT), being deployed under the direction of Sarnoff and several other companies.

II. Standards-Compatible Robustness Approaches

While a great variety of video coding techniques have been developed as a result of research over the past several decades, for the foreseeable future the commercial technology for video coding will be dominated by the ISO (MPEG-1, MPEG-2, MPEG-4) and ITU (H.324, and more specifically H.263 and related video coders) standards which share a common basic approach. These standards combine block-based motion compensation based on one or more nearby frames with discrete cosine transform (DCT) coding of the motion prediction error. To maximize coding efficiency, both the motion compensation information and the transformed prediction error are represented using variable length (Huffman) codes. There are many features of these standards, including the extensive use of variable length codes, that can lead to vulnerability to channel errors.

Clearly, video coding algorithms designed with error robustness as a primary constraint and without the requirement of standards compatibility would use quite a different approach and would get correspondingly better error resilience. Papers published in recent years have examined both standards-compatible and nonstandards-compatible approaches to robust video coding. Examples of techniques include layered source coding, classified bit streams, combined source-channel coding, FEC, ARQ, error concealment, and combinations of the above [1]. While we recognize the importance of research in error-robust, nonstandards compatible video coding, in this paper we emphasize error-robustness enhancements that fall within the framework of existing and emerging standards for wireline video communications. This is motivated by our expectation that both wireline and wireless systems will experience dramatic growth in the coming years, with the result that wireline systems will remain dominant in the general communications infrastructure. Commercial practicability therefore demands that solutions for wireless video be maximally compatible with those used for wireline systems, involving little or no transcoding at the wireline/wireless interface. A similar argument can be made for military systems, which while involving strong differences in the application requirements, still face strong cost pressures to leverage (and possibly enhance) commercial solutions wherever possible.

The constraints imposed by the standards on development of error robust techniques are less restricting than might be expected. One of the important lessons of recent work in MPEG-4 and H.324 is that it is possible to work within the framework of these standards to identify changes that have minimal impact on the complexity and syntax but which lead to important improvements in robustness. In addition to addressing the robustness of the video codec, it is also critical to consider the effects that errors occuring in the multiplexing and packetization layers can have on the encoded video bit stream.

For completeness we give a very brief overview of the video coding standards here, with an emphasis on features relevant to error robustness. Readers interested in more information on the standards are encouraged to refer to the standards documents themselves [2], [3] or to the tutorials and overviews such as those in [4]. Work on video coding standards has proceeded primarily in ISO and ITU. ISO has developed the MPEG-1, MPEG-2, and most recently MPEG-4 standards. Each of these standards is actually an umbrella term for a set of specifications for different aspects of audiovisual compression, including audio coding, video coding, multiplexing, and others. MPEG-1 and MPEG-2 were formally completed several years ago, while work on MPEG-4 is ongoing and anticipated to finish in 1998. While MPEG-4 covers a wide range of multimedia applications, an important aspect of MPEG-4 is focused on low bit-rate video coding, designed with error resilience in mind. Therefore, when discussing ISO we will refer primarily to MPEG-4.

The ITU has developed the specification for H.324 low bit-rate multimedia terminals. H.324 is also an umbrella term, comprising G.723 audio coding, H.223 multiplexing, H.245 control, and a series of video coding standards including H.261, H.263, and most recently, H.263 Version 2 [2], or H.263+ as it is known in the standardization community. When referring to ITU video coding in general we will use the designation H.26X. It should be noted that the H.26X standards are not unique to H.324. For example, the ITU H.323 system specification for packet-switched networks also uses H.26X video coding. The ITU, like ISO, has only begun to consider error robust video in the most recent video coding specification, so the discussion here related to ITU video coding will primarily on H.263+, with some comparative references to H.263.

Until relatively recently, the ISO and ITU video coding efforts were carried out independently. This is partly due to the differences in the charters of the organizations; ISO is charged with developing solutions for storage, while the ITU is concerned with communications. However, the goals of efficient storage and efficient communications are clearly quite closely related, and in the most recent generation of

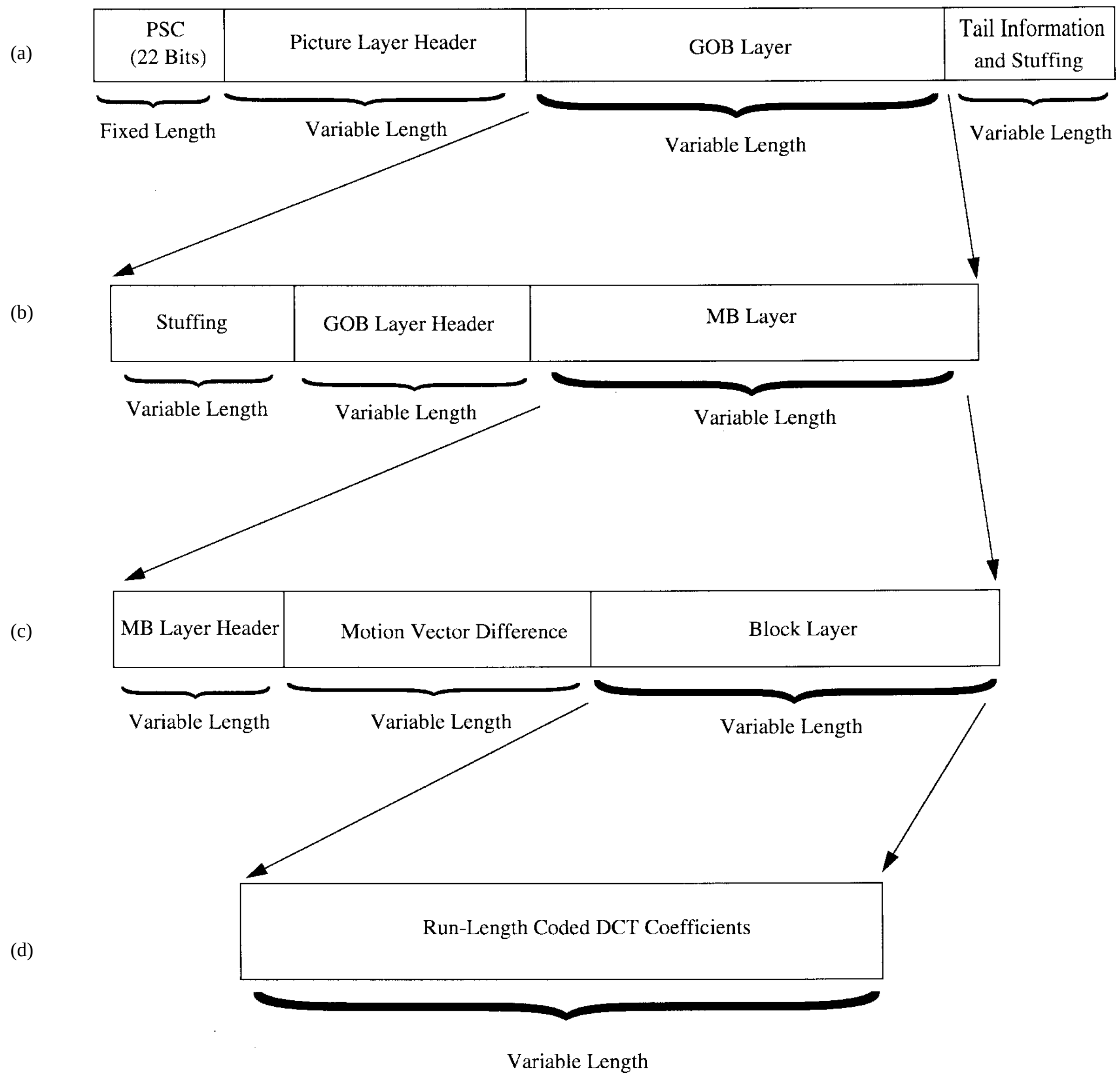


Fig. 1. Basic data hierarchy of H.26X: (a) picture layer; (b) group of blocks (GOB) layer; (c) macroblock (MB) layer; and (d) block layer.

standards (MPEG-4 and H.263+) there has been extensive collaboration between the ISO and ITU efforts, though there remain some differences in details of the video coding algorithms.

Fig. 1 illustrates the data hierarchy and syntax used in H.26X video coders. At the highest level is the picture layer, which begins with a 22-bit picture start code (PSC) followed by header information. Each picture frame is partitioned first into groups of blocks (GOB's), then into macroblocks (MB's) measuring 16 × 16 pixels, and finally into 8 × 8 pixel blocks. There is some potential confusion in the terminology because the GOB consists of one or more rows of MB's, not blocks, so a more apt name for GOB might have been "group of macroblocks." All the information for each block is grouped at one location in the bit stream; for example the motion and DCT data for block N is transmitted before the motion and DCT data for block N+1. To improve efficiency, both the motion vectors the DCT coefficients for intracoded blocks are coded predictively, as opposed to absolutely.

The MPEG-4 video coder can be understood as a generalization of H.263 [3], [4]. MPEG-4 utilizes the idea of "video objects (VO)" which corresponds to entities (e.g., foreground and background) in the bit stream that the user can access and manipulate. The MPEG-4 encoder is composed of two main parts: the shape coder and the traditional motion/DCT coder which is applied to each video object plane (VOP). As in H.263+, the MPEG-4 hierarchy of data includes block (8 × 8), and MB (16 × 16) layers. At the layer corresponding to the GOB layer of H.263+, MPEG-4 has the video object layer (VOL). In contrast with H.263+ GOB's, a VOP in MPEG-4 can be of arbitrary shape and does not have to correspond to an integer number of rows of MB's. When the VOP is rectangular, the shape information is not transmitted. In this case, the MPEG-4 video coding algorithm has a structure very similar to MPEG-1/2 and H.26X.

There are several features of this syntax and coding approach that are of interest from the standpoint of channel errors. First, as mentioned above, the motion vectors, DCT data, and much of the other information is coded using variable length codes, which in general can be desynchronized by errors. Second, the motion and DCT information is coded predictively, which will cause errors to propagate once they have occurred. Third, motion and DCT data are coded together for each block. A more error-resilient approach (and one which is adopted by the error-resilient mode of MPEG-4) is to partition the data so that all motion vector information for each GOB is transmitted first, followed by all of the DCT data. Fourth, the start codes found at the beginning of each picture and of each GOB have advantages and disadvantages when errors are present. On the positive side, they can serve as synchronization markers in the event that a decoder becomes "lost" due to errors in the data. The disadvantage is that the start codes themselves can be corrupted.

III. Error-Resilience Tools for Video

The goal of traditional video coding is to eliminate both spatial and temporal redundancy in the video signal. However, to achieve high video quality for transmission over an error-prone channel, it is highly desirable to have video codecs designed with error resilience in mind ([5]–[8]), and it is sometimes beneficial to preserve some redundancy in the source coding stage intentionally in order to support increased resilience.

In a layered coding approach, essential information for the video source is transmitted in a base layer, which can be used independently to reproduce video signal to an acceptable quality. Supplementary information is transmitted in higher enhancement layers, which, when used with base layer, can improve video quality at the decoder. Layered coding is most effective when the video bit stream is transmitted over channels for which transport prioritization is possible (e.g., ATM networks, in which one bit in the cell header is used to signal its priority) or when the level of error protection applied to the coded video can easily and quickly be altered, such as in the H.223/A mux.

The concept of layered coding is embodied in the MPEG and H.263+ standards through temporal, spatial, and SNR scalability. In temporal scalable coding, the base layer contains a bit stream with a lower frame rate, and the enhancement layers contain information to obtain higher frame rates. In spatial scalable coding, the base layer codes a subsampled version of the input video signal, and the enhancement layers contain information for obtaining higher spatial resolution at the decoder. The coder can also encode the input signal with a coarser quantization, which is then transmitted in the base layer, with finer detail information transmitted in higher enhancement layers. This approach is called SNR scalability. All three types of scalability are standardized in H.263+ in the temporal, SNR, and spatial scalability modes.

The H.263+ and MPEG-4 standards also include options that allow the encoder to produce a bit stream which is slightly less efficient in representing the video, but which is designed to make the task of error concealment at the decoder easier. For example, in the reference picture selection mode of H.263+, it is possible to select the reference picture for motion prediction in order to suppress temporal error propagation due to intercoding. The information which specifies the selected picture for prediction is included in the bit stream. Provided that a back-channel is available, the decoder can tell the the encoder which frames to use for motion compensation, so that any frames that the decoder has identified as being corrupted will not be referenced, thus preventing propagation from motion compensation. When there is no back-channel, the encoder can partition frames into several independent and interleaved groups, or threads, each of which is coded independently without using frames in other threads, so as to make the bit stream more resilient to channels that suffer from both bit errors and packet loss. Because motion vectors predicted using frames that are further apart are usually larger, coding efficiency will be lower because more and longer code words will be used to code long motion vectors obtained. As another example, in the independently segmented decoding mode of H.263+, picture segment boundaries are treated as picture boundaries so that no data dependencies across segment boundaries are allowed. This prevents propagation of errors and enhances error resilience and recovery capabilities at the cost of a slightly lower ability to exploit dependencies across segments.

The MPEG-4 standard is also the first video standardization effort that explicitly included an error resilient mode of operation containing a set of new error resilient video coding tools and ideas. The error resilience tools developed for MPEG-4 can be divided into three classes: 1) error isolation; 2) data recovery; and 3) error concealment.

A. Error Isolation

Error isolation tools, as the name implies, try to prevent error propagation in the bit stream when errors occur. This is often achieved by placing "resynchronization markers" in the compressed bit stream and by using a technique called "data partitioning."

1) Resynchronization Markers: Resynchronization markers are specially designed bit patterns that are usually placed at approximately regular intervals in the video bit stream. The function of these markers is to divide the compressed video bit stream into segments that are as independent of each other as possible. By searching for these markers, the decoder can reliably locate each segment without actually decoding the packet, and thereby prevent error propagation across different segments separated by markers. Each data segment of the bit stream should generally contain one or several complete logical entities of video information (i.e., blocks, MB's, etc.) so that the decrease in coding efficiency

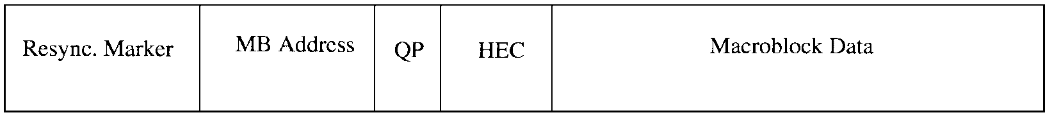


Fig. 2. Error-resilient video packet.

due to not exploiting dependencies between segments can be minimized. The length of each segment is usually chosen to achieve a good tradeoff between the overhead introduced by the markers, and reliability of the detection of markers when errors occur.

One of the resynchronization approaches adopted by MPEG-4, referred to as the packet approach, is similar to the GOB structure utilized by the H.26X standards. The GOB header contains a GOB start code which is different from a picture start code, and contains information which allows the decoding process to be restarted (i.e., resynchronize the decoder to the bit stream and reset all coded data that have been predicted). The GOB approach to resynchronization is based on spatial resynchronization. That is, once a particular MB location is reached in the encoding process, a resynchronization marker is inserted into the bit stream. A potential problem with this approach is that since the encoding process is variable rate, these resynchronization markers will most likely be unevenly spaced throughout the bit stream. Therefore, certain portions of the scene, such as high motion areas, will be more susceptible to errors, which will also be more difficult to conceal. By contrast, the video packet approach adopted by MPEG-4, is based on providing periodic resynchronization markers throughout the bit stream. In other words, the length of the video packets are not based on the number of MB's, but instead on the number of bits contained in that packet. If the number of bits contained in the current video packet exceeds a predetermined threshold, then a new video packet is created at the start of the next MB.

Fig. 2 shows a typical video packet in MPEG-4. The resynchronization marker placed at the start of a new video packet is distinguishable from all possible VLC code words as well as the VOP start code. Header information is also provided at the start of a video packet. This header contains the information necessary to restart the decoding process, including the macroblock address of the first macroblock contained in this packet and the quantization parameter (QP) necessary to decode that first MB. The MB number provides the necessary spatial resynchronization while the quantization parameter allows the differential decoding process to be resynchronized. Following the QP is the header extension code (HEC). As the name implies, the HEC is a single bit to indicate whether additional information will be available in this header. If the HEC is equal to one then the following additional information is available in this packet header: modulo time base; temporal reference; VOP prediction type.

Utilizing the error-resilience tools within MPEG-4 can involve some small sacrifices in coding efficiency. For example, all predictively encoded information must be confined within a video packet to prevent the propagation of errors caused by predictive coding/decoding steps in the algorithm. In addition to the GOB approach and video packet approach to resynchronization, a third method called fixed interval synchronization has also been adopted by MPEG-4: fixed interval synchronization. This method requires that VOP start codes and resynchronization markers (i.e., the start of a video packet) appear only at allowable, fixed interval locations in the bit stream. This helps to avoid the problems associated with start code emulations. Although errors can cause emulation of a VOP start code, this emulation will only be problematic in the unlikely event that it occurs at a location permitting GOB start codes.

2) Data Partitioning: In the absence of any other error-resilience tools, the data between the synchronization point prior to the error and the first point where synchronization is re-established is discarded when errors are detected in the decoding of "real" data. If the resynchronization approach is effective at determining the amount of data discarded by the decoder, then the ability of other types of tools which recover data and/or conceal the effects of errors is greatly enhanced.

To achieve better error isolation in the video packet and fixed interval synchronization approaches, MPEG-4 introduced data partitioning to further improve the ability of the decoder to localize an error. When the data partitioning syntax is used, video bit stream between two consecutive resynchronization markers (often called a "packet") is divided into finer logic units. Each logic unit contains one type of information (e.g., DCT) for all the MB's in the whole packet (when present, shape data are also partitioned). This is in contrast to the nondata-partitioned syntax, in which each MB contains its own header, motion, and texture data. For the decoder to locate each logic unit, secondary markers are placed between logic units. Unlike the resynchronization marker, which needs to be free of emulation from header, motion, and DCT data, these secondary markers need only to be free from emulation by data in the logic units that immediately proceed them. For example, the marker between motion and DCT data needs only to be free from emulation by motion data; it can be emulated by DCT data.

When the decoder detects an error in a packet using the data partitioning syntax, it can then search for the next secondary marker in the packet and start decoding the next logic unit within the same packet. Because the decoder only needs to discard the rest of the logic unit, instead of the rest of the packet, more data can be salvaged and utilized. Without data partitioning, the decoder would need to compensate for the lost of header and motion and DCT data for all macroblocks from the one in which the error

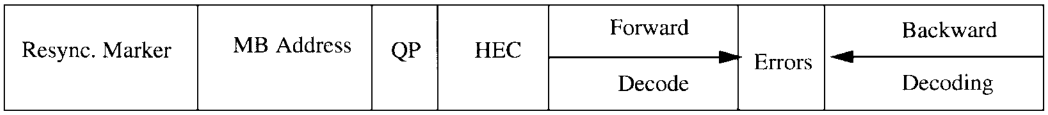


Fig. 3. Error localization using reversible VLC's.

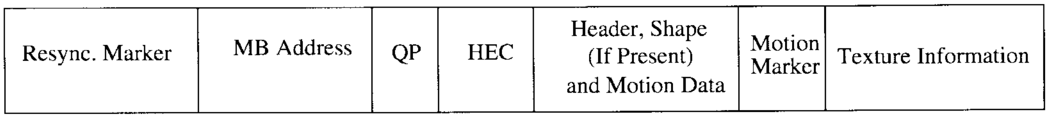


Fig. 4. Error concealment with data partitioning.

is detected. When data partitioning is used, each correctly decoded logic unit contains one type of information for all MB's in the packet, the task of error concealment is thus made much easier.

B. Data Recovery

After synchronization has been re-established, data recovery tools attempt to recover data that would otherwise be lost. One of the most important data recovery tools for video, and one that has been adopted in both MPEG-4 and H.263+, is reversible variable length codes (RVLC). In this approach, the variable length code words are designed such that they can be read both in the forward as well as the reverse direction. Intelligently designed RVLC's and corresponding decoding methods can significantly improve the error robustness of the bit stream, with little or no loss of coding efficiency ([9], [10]).

An example illustrating the use of an RVLC is given in Fig. 3. In general, when a burst of errors has corrupted a portion of the data, all data between the two synchronization points would be lost. However, as shown in Fig. 3, an RVLC enables some of that data to be recovered. By providing the capability of cross checking between the output of the forward and backward decoder, at a modest cost in increased complexity, RVLC's can also help the decoder to detect errors that are not detectable when nonreversible VLC's are used, or provide more information on the position of the errors, and thus decrease the amount of data unnecessarily discarded.

To fully utilize the error localization properties of the RVLC's, the syntax for the MB layer needs to be modified in order to group all data coded with one RVLC table together. This is necessary to ensure that the reverse decoding operation will not be blocked by a nonreversible code word or reversible code words from another table. By grouping code words for the same type of information (e.g., motion, DCT) for all the MB's in a packet together and placing markers between different logic units, data partitioning provides the necessary syntax change for the applications of RVLC's, and thus is often used in conjunction with RVLC. Fig. 4 illustrates the syntactic structure of the data partitioning mode. Like the use of RVLC's, the use of the data partitioning syntax is also signaled to the decoder in the VOL layer.

It should be noted that data partitioning alone can be applied without RVLC's. However, using RVLC's for the coding of each logic unit will maximize the benefits of the data partitioning syntax with little or no extra overhead.

C. Error Concealment

Using *a priori* knowledge about image/video signals, it is possible to include "error concealment" capabilities in decoders so that the severity of artifacts resulting from transmission errors is minimized. Error concealment is an extremely important component of any error robust video codec. Spatial and temporal interpolations are often utilized in error concealment methods. Examples include maximally smooth recovery [11], projection onto convex sets, [12], and various motion vector and coding mode recovery methods such as motion compensated temporal prediction [13]. Like the error-resilience tools discussed above, the effectiveness of an error-concealment strategy is highly dependent on the performance of the resynchronization scheme. If the resynchronization method can accurately localize the error, then the error concealment problem becomes much more tractable. Simple concealment strategies based on copying blocks from previous frames instead of displaying corrupted blocks from a current frame can be very effective.

Error detection and localization are usually achieved by checking if the information decoded is "legal" given the syntax of the bit stream. When RVLC's are used, the decoder has the additional capability of error detection by cross checking of the forward and backward decoded results. A more extensive discussion of error concealment is contained in several of the other papers in this special issue.

D. Evaluation Procedures

Performing an objective evaluation of the merits of various robustness techniques is a challenging task. There are clearly many different types of errors that can be applied to a coded video bit stream, and no one set of robust coding approaches will perform optimally across all error-prone channels. The most thorough framework constructed to date for this task is the algorithm evaluation procedure developed by the the MPEG-4 *ad hoc* group on error resilience. In the core experiments defined by this group, errors are applied to the bit stream using software provided by NTT DoCoMo. There is a 1.5-s period of error-free transmission at the beginning of the bit stream, after which the channel becomes noisy. The NTT DoCoMo software can simulate random error channels, packet-lossy channels, and bursty channels. The statistics measured on the decoded video include the following.

1) Peak-signal-to-noise ratio (PSNR).
2) Fraction of bits received in error (pbd), e.g., the ratio of the total number of bits discarded by the decoder to the total number of bits transmitted.
3) Fraction of frames received in error (pfd), defined as ratio of the total number of frames discard at the decoder and the total number of frames transmitted.
4) Overhead, defined as the ratio of the total number of additional bits introduced for error resilience, as compared to the nonerror-resilient mode, to the total number of bits transmitted with error resilience.

To assess correctly the effectiveness of error-resilient algorithms, only the simplest error "concealment" methods (such as copy from previous decoded blocks or frames) are used in the core experiments. The purpose of using error concealment is to prevent the decoder from "crashing" in the presence of errors, and to collect enough data for algorithm evaluation.

IV. Robust Multiplexing

In experiments to explore video coding error robustness, error patterns derived from channel models are often applied directly to an encoded video bit stream, which is in turn sent to the input of a video decoder. While this approach can be very useful for exploring the value of different codec design approaches, it does not constitute a realistic model for a full end-to-end video communications system. For those networks that use protocols at other layers to ensure that data (in this case, video) is delivered error free, application of channel errors directly to the compressed video is unreasonably pessimistic. For those networks that do not use retransmission, it fails to account for any forward error correction performed at other network layers, and more critically, it fails to account for the multiplexing and packetization steps. The multiplexing and packetization can constitute an important source of error because of the possibility that video can be misdelivered, causing large chunks of data to disappear from the video bit stream seen by the receiver.

Probably the most extensive effort to jointly consider multiplexing and video coding has been performed by the ITU during development of H.324. The H.324 suite of specifications includes the H.223 multiplexer, which is designed to support multiplexing of data from multiple sources on a circuit-switched network. While the original H.223 specification targeted the V.34 modem and was therefore designed with relatively low error rates in mind, interest in using H.324 over wireless channels led to work to extend H.223 to allow operation over error-prone channels. This work, which was carried out in large part during the period 1995–1997, led to the development of a series of annexes to H.223. With the addition of these annexes, H.223 now offers a hierarchical, multilevel multiplexing structure, allowing implementers to trade off robustness against overhead and complexity.

H.223 is a connection-oriented multiplexer that combines data sources into a single bit stream. In the simplest default layer of H.223 (level 0), packets are variable length and are delimited by an 8-bit synchronization flag. A synchronization flag is followed by an 8-bit header that identifies the contents of the packet and then by the payload, which in general can contain a mix of various sources. The end of the packet is indicated by the next appearance of the 8-bit synchronization flag, Bit stuffing is performed on all data between synchronization flags to avoid flag emulation. Fig. 5(a) illustrates the H.223 Level 0 packet structure. The principal vulnerabilities of H.223 Level 0 lie in the bit stuffing, and in the short, and therefore vulnerable synchronization flags and headers.

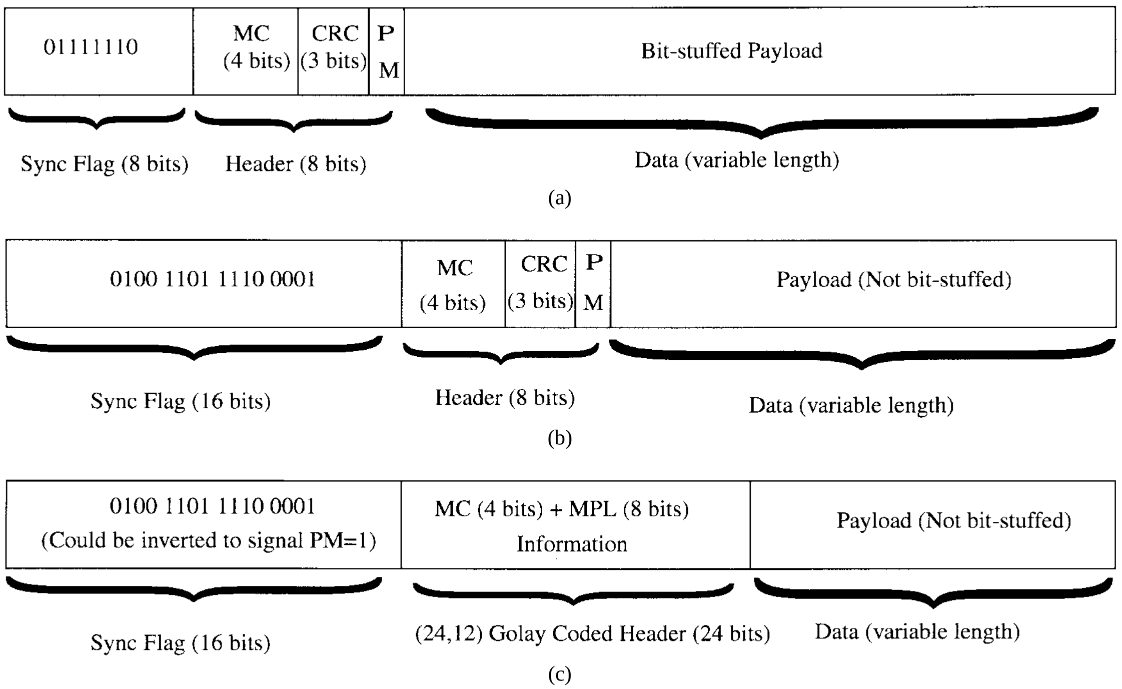
In level 1 [Fig. 5(b)] bit stuffing is not performed, and a longer synchronization flag is used. The flag can be emulated by the data, but such emulations are not usually problematic. In level 2 [Fig. 5(c)] further robustness is enabled by lengthening and adding error protection to the header that describes the contents of the packets.

Table 1 provides some information on the performance of these different levels. The table considers the ability of the H.223 multiplexer levels to deliver packets over three different Rayleigh channels. For each channel and multiplexer, the table provides information on the percentage of packets that are correctly delivered (e.g., with no errors), the number of packets that are delivered with undetected errors, and the throughput in terms of bits. As expected, the more robust multiplexer levels lead to improved communication. The degree of improvement is greater for poorer channels. Among the categories considered, the most important improvement as the multiplexer level is increased is in the percentage of data delivered to the decoder that is corrupted (Y_2/X in the table). Significantly, the most robust level of the multiplexer (level 2) reduces the amount of corrupted data by over an order of magnitude.

In more general terms, the most important message in Table 1 is that channel error-induced failures at other network layers are likely to have extremely important consequences at the layers where the source codecs (in particular the video codec) lie. For example, it is quite unlikely that an H.324 system designed for wireline environments (and therefore using H.223 level 0) would function over the channels which would cause several percent of the packets delivered to the video decoder to contain significant numbers of errors. Even a video decoder modified to be extremely robust would be of only marginal use in a system in which a few percent of the video bits are misdelivered (to an audio decoder for example), leading to large gaps in the received bit stream seen by the video decoder. Designers of video systems for wireless environments will have to take a system level view to ensure that a consistent level of robustness is maintained across the multiplexing and video subsystems.

V. Wireless Video Testbeds

In the recent few years there has been a growing set of testbeds developed to explore the issues of robust wireless



(a)

(b)

(c)

Fig. 5. H.223 Packet structure: (a) layer 0; (b) layer 1; and (c) layer 2. MC: multiplex code. PM: packet marker. MPL: multiplex payload length.

Table 1
Performance Comparison of H.223 Levels

Channel Model	BER	Level	Y_1/X	Y_2/X	Y_1+Y_2	Throughput
Ray0_14	$9.3x10^{-3}$	0	90.33%	3.57%	93894	89.57%
		1	92.02%	2.65%	94672	91.30%
		2	92.93%	0.07%	93112	93.14%
Ray0_18	$3.7x10^{-3}$	0	94.97%	2.51%	97480	94.37%
		1	95.83%	2.02%	97856	95.27%
		2	96.03%	0.07%	96217	96.07%
Ray0_22	$1.5x10^{-3}$	0	97.72%	1.30%	99025	97.39%
		1	98.74%	0.47%	99212	98.64%
		2	97.42%	0.07%	97625	97.45%

Note: In the above table, X is the total number of packets transmitted (set to 10000 in the simulations), $Y=Y_1+Y_2$ is the total number of packets received. The quantity Y_1 is the number of packets received correctly, Y_2 the received with undetected errors (e.g. if a flag is missed, resulting in an extra long packet, this is count in Y_2 . Other events that count in Y_2 include detecting a flase flag, resulting in an artificially short packet, or the failure of the header code to detect a correuption in the Multiplex code. In this case, an attempt is made to demultiplex using the wrong Multiplex Code). Throughput in the table is expressed as the ratio of the total number of bits in the Y packets the decoder received over the total number of bits in the X packets that the sender generated.

video communications. We describe here two example systems that one of the authors was involved in developing and experimenting.

A. DARPA Wireless Interworking Testbed (WIT)—Low Bit-Rate Video Coding and Transmission

In June 1996, a consortium consisting of Sarnoff, Lucent Technologies, Bellcore, and the U.S. Army CECOM collaborated in developing a wireless testbed which can be used to test the performance and characteristics of data, image, and video in a mixed network environment and conditions. The program, supported by Technology Reinvestment Program (TRP), constructed a heterogeneous testbed of wireless and wireline components to allow interoperability testing of emerging commercial and government market information devices and systems. Using this network Sarnoff recently demonstrated low bit rate scalable video system over a mixed wireless/wireline network.

Sarnoff passed 45-kbit/s, 30-frames/s video using the H.263 codec with synchronized GSM audio. The image at receiving side was a QCIF 30 frames/s picture with lip synchronized audio. The multimedia application was embedded in an RTP enabled IP stack and was transported across the network using UDP on ATM.

Presently this demonstration is being instrumented to act as an application for the testing of wireless and wireline internet audio/video subsystems and systems.

B. Handheld Multimedia Terminal

The Handheld Multimedia Terminal (HMT) is a new generation wireless radio system which incorporates advanced communications capabilities, high-performance computing, and state-of-the-art video and imagery compression technologies. The HMT is being developed, with partial support from the Defense Research Projects Agency (DARPA), by a consortium composed of ITT, Honeywell, Sarnoff Corporation, and Medical Communications Systems. The HMT is designed to function in both military and high multipath, commercial communications environments. It will also provide reliable communications within building environments such as hospitals. The radio communications operate in a tetherless mode with over 1 Mbit/s of bandwidth, which is used for point-to-point, and automatic relay of communications for terminals, which cannot communicate directly. Novel media access and transport communications protocols have been developed to allow reliable, efficient communications over the shared bandwidth. The terminal incorporates a Pentium class processor running Windows 95. An MPEG-4 compliant codec is used to support collaborative multimedia communications among terminal users. The architecture incorporates standard Ethernet communications so that the HMT can be easily interconnected with other networks. Initial military markets will support for the Army for communications among dismounted soldiers, ground vehicles, and rotary wing aircraft. Commercial markets include process control environments, e.g., refineries and chemical processing facilities, law enforcement applications, and support for medical personnel within hospital environments. The HMT incorporates two advanced multimedia capabilities in addition to its advanced radio and high-performance processing features. Still image and graphics are compressed with the MPEG-4 still texture coding tool, Multiscan Zero Tree Entropy (MZTE) compression. The HMT will be one of the first products to incorporate this high-performance wavelet-based compression technology for image and graphics transmission, coupled with an MPEG-4 compliant codec. The MPEG-4 compliant codec implements Sarnoff-proprietary fast motion estimation scheme, scalable rate control, and error-resilence tools.

VI. Conclusions

With the growth in wireless bandwidth, the increasing availability of low-power processing, and the market pressure from increasing functionality in wireline systems, it is only a matter of time before wireless exchange of imagery becomes commonplace. To best meet the technical challenges that wireless video offers, researchers need to continue to explore both standards-compatible and nonstandards-compatible approaches to wireless video and to ensure that the best of the techniques that result migrate quickly to the commercial world.

In addition to contributing to the standards development process, researchers in the field of wireless video can make substantial contributions to implementation strategies. Since the video coding standards only specify the contents of an uncorrupted coded video bit stream, it is quite possible, and in fact very common, to build a video decoder which is standards compatible but extremely fragile. Though robust implementations have not generally been sought in the past because most video communications have used very reliable communications environments, the next few years are certain to see a very large growth in commercial and academic work in these areas.

Acknowledgment

The authors of the paper wish to acknowledge Dr. J. Brailean from PacketVideo Technology, Dr. W. Stephens, and P. Hashfiled for their input and contribution on MPEG-4 error resilience, WIT, and HMT testbeds.

References

[1] Y. Wang and Q. Zhu, “Error control and concealment for video communication: A review,” *Proc. IEEE*, vol. 86, pp. 974–997, May 1998.

[2] *Video Coding for Low Bitrate Communication*, ITU-T Recommendation H.263 Version 2, Jan. 1998.

[3] *Coding of Moving Pictures and Associated Audio Information*, ISO/IEC JTC/SC29/WG11, 1997.

[4] *IEEE Trans. Circuits Syst. Video Technol. (Special Issue on MPEG-4)*, vol. 7, Feb. 1997.

[5] Y.-Q. Zhang, Y.-J. Liu, and R. Pickholtz, “Layered image transmission over cellular radio channels,” *IEEE Trans. Veh. Technol.*, vol. 43, Aug. 1994.

[6] R. Stedman, H. Gharavi, L. Hanzo, and R. Steele, “Transmission of subband-coded images via mobile channels,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, Feb. 1993.

[7] Y.-Q. Zhang and X. Lee, “Performance of MPEG codecs in the presence of errors,” *J. Visual Commun. Image Representation*, vol. 5, no. 4, pp. 379–387, Dec. 1994.

[8] *IEEE Trans. Circuits Syst. Video Technol. (Special Issue on Wireless Video)*, vol. 6, Apr. 1996.

[9] J. Wen and J.D. Villasenor, “A class of reversible variable length codes for robust image and video coding,” *Proc. 1997 IEEE Int. Conf. Image Processing*, vol. 2, Santa Barbara, CA., Oct. 1997, pp. 65–68.

[10] *Description of Error Resilient Core Experiments*, ISO/IEC JTC1/SC29/WG11 N1383, Nov. 1996.

[11] Y. Wang, Q. Zhu, and L. Shaw, “Maximally smooth image recovery in transform coding,” *IEEE Trans. Commun.*, vol. 41, pp. 1544–1551, Oct. 1993.

[12] H. Sun and W. Kwok, “Concealment of damaged block transform coded images using projection onto convex sets,” *IEEE Trans. Image Processing*, vol. 4, pp. 470–477, Apr. 1995.

[13] M. Ghanbari, “Cell-loss concealment in ATM video codes,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 238–247, June 1993.

John D. Villasenor (Senior Member, IEEE) received the B.S. degree from the University of Virginia, Charlottesville, in 1985 and the M.S. and Ph.D. degrees from Stanford University, Stanford, CA, in 1986 and 1989, respectively, all in electrical engineering.

From 1990 to 1992, he was with the Radar Science and Engineering Section of the Jet Propulsion Laboratory, Pasadena, CA. He joined the University of California, Los Angeles, in 1992 and is currently Professor and Vice Chair of the Electrical Engineering Department. His research interests include source and channel coding, robust communications, and configurable computing.

Ya-Qin Zhang (Fellow, IEEE) was born in Taiyuan, China, in 1966. He received the B.S. and M.S. degrees in electrical engineering from the University of Science and Technology of China (USTC) in 1983 and 1985, respectively, and the Ph.D degree in electrical engineering from George Washington University, Washington, DC, in 1989.

He joined Microsoft Research China, Beijing, in January 1999. He was previously the Director of Multimedia Technology Laboratory at Sarnoff Corporation, Princeton, NJ (formerly David Sarnoff Research Center and RCA Laboratories). His laboratory is a world leader in MPEG-2/DTV, MPEG-4/VLBR, and multimedia information technologies. He was with GTE Laboratories, Inc., Waltham, MA, and Contel Technology Center, Virginia, from 1989 to 1994. He has authored and coauthored over 150 refereed papers and 30 U.S. patents granted or pending in digital video, internet, multimedia, wireless, and satellite communications. He has been an active contributor to the ISO/MPEG and ITU standardization efforts in digital video and multimedia. Many of the technologies he and his team developed have become the basis for start-up ventures, commercial products, and international standards.

Dr. Zhang has received numerous awards, including several industry technical achievement awards and IEEE awards. He received the Research Engineer of the Year Award in 1998 from Central Jersey Engineering. He recently received the National Outstanding Young Electrical Engineer of 1998 Award from Eta Kappa Nu. He was Editor-in-Chief of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and he was a Guest Editor for the PROCEEDINGS OF THE IEEE February 1995 Special Issue on Advances in Image and Video Compression. He serves on the editorial boards of seven other professional journals and over 12 conference committees.

Jiangtao Wen (Associate Member, IEEE) received the B.S., M.S., and Ph.D. degrees (with honors) from Tsinghua University, Beijing, China, in 1992, 1994, and 1996, respectively, all in electrical engineering.

From 1996 to 1998, he was a Researcher in the Electrical Engineering Department at the University of California, Los Angeles. He joined PacketVideo Technology, San Diego, CA, in 1998. His current research interests include wireless multimedia communications and source and channel coding. He has been actively involved in the standardization efforts of the ITU-T (H.263) and ISO (MPEG-4) for wireless multimedia communications.