ERROR RESILIENCE AND CONCEALMENT IN VIDEO CODING

A. K. Katsaggelos, F. Ishtiaq, L. P. Kondi, M.-C. Hong, M. Banham[+], *J. Brailean*[+]
Department of Electrical and Computer Engineering, Northwestern University,
Evanston, IL 60208-3118
[+] Motorola Corporate Systems Research Laboratories
Schaumburg, IL 60196-1078
e-mail: {aggk,faisal,lkon,hong}@ece.nwu.edu,
{banham,brailean}@ccrl.mot.com

ABSTRACT

In this paper we review error resilience and concealment techniques which have been developed both within and outside the various videoconferencing standards. We consider the H.324 videoconferencing standard with its accompanying lower-level H.263 and H.223 video and multiplexing standards, MPEG-2, and MPEG-4. We then describe an error resilience algorithm commonly used with variable length codewords, and finally review error concealment techniques that have appeared in the literature.

1 INTRODUCTION

Video compression is the enabling technology behind the multimedia revolution as represented by digital storage media, broadcasting, Internet video, mobile communications, mobile video surveillance, and command and control. A large number of video coding algorithms have appeared in the literature [1]. In parallel a number of video coding standards have emerged, or are currently under development. Associated with the video coding standards are a number of systems and multiplex standards. It is often the system layer which ultimately determines the performance of video communications in the presence of errors.

Transmitting video in the presence of errors is not a novel concept by any means. For decades, analog NTSC and PAL video has been modulated and transmitted over noisy broadcast channels, often with significant degradation to video quality. For analog broadcast applications, however, the video quality typically degrades gracefully as a function of the receiver's distance from the transmitter. In the world of digital video, an entirely different situation arises. Compressed digital video is more susceptible to the effects of channel noise, when bit errors are not entirely removed by error correction. Because information is typically predictively coded, errors can propagate through a decoded sequence over time, making it difficult to hide the errors from the viewer.

In this paper we review in Sec. 2 video compression and transmission standards, following [2]. The error-resilient entropy code (EREC) is reviewed in Sec. 3. Error concealment algorithms are reviewed in Sec. 4, and a technique for recovering lost motion vectors is described and demonstrated.

2 VIDEO COMPRESSION AND TRANSMISSION STANDARDS

Since the late 1980's, a number of efforts have been made towards developing standards for video coding and more generally for video conferencing and multimedia transmission over circuit and packet switched networks. These standardization efforts have been carried out by two separate international bodies, the International Telecommunications Union/Telecommunications Standardization Sector (ITU-T) and the International Organization for Standardization/International Electrotechnical Commission (ISO/IEC). The cooperation between these bodies and various international organizations, research institutions, universities, and companies have resulted in the ITU-T Series H Recommendations [3, 4, 5] and the ISO/IEC MPEG-1, MPEG-2, and the soon to be released MPEG-4 standards for video coding [6, 7]. These standards have become an important and integral part of video coding applications and have spurned may research efforts into the rate control, error resilience, and error concealment aspects of the standards. In this paper we concentrate on the H.324, MPEG-2, and MPEG-4 standards when considering error resilience and concealment. We also focus on the system and multiplex layers since they are an integral part in the performance of video communications systems over error-prone channels.

2.1 ITU-T H.320 H.323 H.324

The ITU H.320 and H.324 videoconferencing standards provide protocols for videoconferencing over circuit-switched networks. H.323 is a standard developed in a similar manner to H.324 that is primarily focused at packet-switched networks. These are all high-level standards that themselves recommend lower layer protocols to address the details of the various aspects such as control, multiplexing, and source coding.

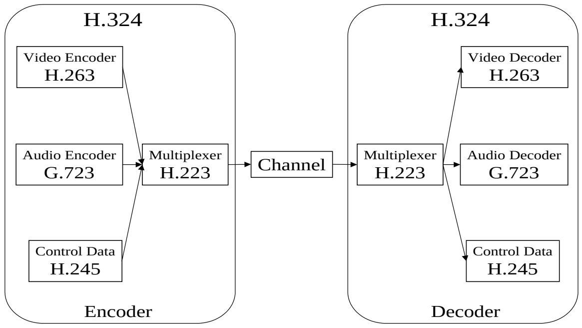

Figure 1: A typical H.324 videoconferencing system.

H.320 was the first generation videoconferencing standard and is the most widely used videoconferencing standard over circuit-switched networks today. It recommends the H.261 and H.263 standards for video coding, G.723 for audio coding, H.245 for control, and H.221 for multiplexing. The more recent standard for videoconferencing is the H.324 Recommendation. It is designed for wireline communications for audio, video, and data over the POTS telephone networks. As with its predecessor, H.324 specifies a host of other standards to handle the various aspects of videoconferencing. It recommends H.261 and H.263 for video coding, G.723 for audio coding, H.245 for control, and H.223 for multiplexing [3]. Figure 1 details the overall structure of a typical H.324 videoconferencing system.

2.2 Multiplexing Layer

Given any videoconferencing system, video and audio data must be synchronized for any meaningful communications to take place. Thus audio and video data must be multiplexed before transmission. The role of the multiplexer is to take the individual compressed audio and video bitstreams and multiplex them for transmission over the channel. A multiplexer can also add additional control data at this point before transmission. It is this multiplexed data that will be exposed to the channel and hence to channel errors, thus the need for error correction at this level is crucial.

2.2.1 Variable vs. Fixed Length Packets

There are two ways in which packets can be structured: fixed length and variable length packets. Using fixed length packets results in a good synchronization between the encoder and the decoder since the length of the incoming data packets is known beforehand. This, however, can be quite inefficient if the source data is not readily available from the video codec and the multiplexer will be forced to add stuffing bits to ensure that the packets are all of equal size. This results in a waste of bandwidth which often times is a critical resource. To avoid having to use stuffing bits, one can use the alternative method of variable-length packets. Unfortunately, the price paid is reduced error resilience because variable length packets must now define their boundaries themselves for the demultiplexer. A loss of this overhead information can cause the demultiplexer to lose synchronization and therefore packets until resynchronization can be reestablished. However, in general, variable length packets offer more coding flexibility with better efficiency [2].

2.2.2 Errors at the Multiplexing Layer

Since the multiplex layer carries the video, audio, and control data, it is very important that it has error resilience. Errors in the multiplexer can occur in several ways and can affect the mux header and/or the payload in catastrophic ways. An error in the mux packet header may cause the audio data to appear as video data and vice versa. This will cause the demultiplexer to send the data to the wrong decoder. If this is done, the incorrect packet will critically impact the quality of the decoded audio or video. A second type of error introduced to the multiplexer packet may be in the form of errors within the packet payload itself. If the demultiplexer is able to pass this data to the audio or video decoders, it is then left to the individual decoders to handle these errors. The decoders can simply discard the entire data, or attempt to correct, contain, and conceal the errors if they can be effectively detected.

2.2.3 H.223 Multiplexing Protocol

Within the H.324 standard, the H.223 standard is the recommended protocol for the multiplexer. H.223 is based on variable length packets with each packet being delimited by a special flag. In its standard mode of operation H.223 is not very robust to errors [2]. Errors in the packet header can lead to the loss of an entire packet and/or to the loss of synchronization. To address the use of H.223 over error-prone channels, three Annexes with increasing levels of protection are provided. These levels, labeled Level 1 to Level 3, provide the multiplexer with increasing levels of robust operation depending of the channel quality.

1. Level 1 dictates the use of a 16 bit pseudo-noise synchronization flag instead of the 8 bit special flag to delimit packets. This makes the detection of a packet easier by adding an extra level of resilience over the base mode of operation.

2. Level 2 adds extra protection to the mux packet header itself by appending additional information into the header and protecting the header with a (24,12,8) Extended Golay Code.

3. Level 3 is the highest level of protection provided by the H.223 Recommendation. It addresses the protection of the mux payload by providing for error detection, interleaving, and ARQ Type I and II retransmission.

These annexes allow for the level of error resilience to be adjusted according to the severity of the channel errors. Furthermore, if Annex C of H.324 is invoked, H.324 can itself dictate to H.223 the level of resilience that the H.223 multiplexer should use and can dynamically adjust these levels during transmission. This makes the error resilience of the system and multiplexer layers adaptive to the channel conditions.

2.3 Video Coding Standards

The H.261, H.263, MPEG-1, MPEG-2, and MPEG-4 standards are all based on hybrid Motion-Compensated DCT (MC-DCT) coding algorithms. They work on the premise of first removing the temporal correlation and then the spatial correlation. Therefore, the compressed data are considerably more important than the original data. Single bit errors in the compressed information can result in enormous errors in the decoded video. Furthermore, due to the predictive nature of video coding as defined by all of the standards, errors will propagate from one frame to the next until an uncorrupted reference is reestablished. For these reasons a number of provisions have been made in the standards to make them more resilient to errors. We shall now discuss the nature of the video coding standards as they pertain to error resilience.

Before we consider each of the standards individually, let us consider certain general aspects of video coding in the presence of errors by looking at the structure of the typical video coding standard and its response to errors.

- Predictive coding and error propagation - The majority of frames encoded by a typical H.261, H.263, MPEG-1, MPEG-2, and MPEG-4 are Inter, or predicted (P), frames. These P frames are encoded as the difference between the motion- compensated previously decoded frame and the source frame. Thus an error in the previous decoded frame will result in the current frame not being decoded correctly. This error in the decoding will continue to propagate until Intra (I) data is available. To deal with this error propagation I macroblocks and I frames can be used to localized the propagation to a small temporal area.

- Motion Vector and Quantization Parameter coding - The motion vectors used for motion compensation are encoded differentially in the video coding standards. This is also the case with the quantization parameter for each macroblock. Since error resilience is weakened by any type of differential coding technique, the corruption of the motion vectors or the quantization parameter can lead not only to temporal corruption of the decoded frames but also to spatial degradation. It is for this reason that the standards have certain modes that facilitate the partitioning of the data according to importance.

- VLC Coding and Resynchronization - The last piece of information to be transmitted in a macroblock are the Discrete Cosine Transform (DCT) Variable Length Codewords (VLC). At the decoder, as the bits for the VLC codeword arrive they are continued to be read in until a valid codeword is recognized. Thus if the bits of the codewords are corrupted, the decoder will not be able to recognize the transmitted codeword and synchronization with the encoder will be lost. The decoder may continue to read in bits until a valid resynchronization marker is encountered at which point the previously read data may be discarded. Errors may never be detected until a start code or a resynchronization code is detected. The use of resynchronization information dispersed throughout a frame can help to localize this loss of synchronization.

As we see from the above error profile of a typical video coder, even minimal errors can impact the decoded video quality greatly. For these reasons all of the video coding standards have made provisions for attempting to minimize and localize the errors. We will now consider the more recent coding standards, namely H.263, MPEG-2, and MPEG-4.

2.4 H.263

H.263 is the second generation video coding standard following H.261. It provides a number of improvements over H.261 including a more powerful 1/2 pel motion estimation and compensation algorithm, overlapped block motion compensation, and the use of SNR, temporal and spatial scalability. Among its error resilience characteristics is the use of the Group of Blocks (GOB) from H.261 as well as a number of Annexes for error resilience.

Group of Blocks (GOB) - A Group of Blocks is defined as a spatially localized group of macroblocks within a frame that is coded independent of one another. By using the GOB structure, errors that would otherwise have propagated to the end of the frame are limited to only those macroblocks within the same GOB. A GOB in a QCIF frame is defined as an entire row of macroblocks, with the GOB header being placed at the left-most macroblock. Figure 2 shows and example of a GOB header appearing in white at the left-most macroblock within a QCIF Inter, or P, frame encoded at 48 Kbps.

Annex H - Forward Error Correction for Coded Video Signal - This Annex allows for 492 bits of coded data to be appended with 2 bits of framing information and 18 bits of parity information to form a frame. Furthermore the framing information bits are such that over eight frames these bits form a specific frame alignment pattern. The forward error correction is via a (511,493) BCH forward error correcting code. This adds Forward Error Correction capabilities to the encoder.

Annex K - Slice Structured Mode - Similar to the GOB structure, the slice structured mode allows for resynchronization information to be added within the frame. However instead of the resynchronization points being spatially determined according to the macroblock number, the slice structure mode places resync points according to the number of bits. This allows for placement of a resynchronization marker at the beginning of any macroblock within the frame and not at the beginning of the GOBs. Each slice is an independently coded unit such that errors will not carry over from one slice to the next. Figure 2 along with GOB headers also shows the locations of slice headers in gray placed every 512 encoded bits.

Annex N - Reference Picture Selection Mode - This annex allows a decoder to signal to the encoder via a backchannel that an error has occurred. The encoder can then be instructed to change the frame used for the prediction of the current frame to the last correctly decoded frame. By doing this, the decoder is able to stop any error propagation that would normally have resulted.

Annex O - Temporal, SNR, and Spatial Scalability Mode - Although Annex O was not designed with error resilience in mind, scalability lends itself to error resilience. This Annex defines independent layers that offer increasing levels of either SNR, temporal, or spatial quality. Thus given a "base" layer with a baseline quality level, independent enhancement layers can be added to enhance the quality. This concept is well suited for error resilience, where the base layer can be a well protected bitstream providing a baseline quality of service, while enhancement layers can be less protected since they will only serve to enhance this quality of service. The ideas of unequal error protection apply very naturally to a scalable bitstream.

Annex R - Independent Segment Decoding(ISD) Mode - The ISD mode of H.263 provides a mechanism to decode a picture with no dependencies across slice or GOB boundaries having non-empty GOB headers. Thus corrupted data cannot propagate across the spatial boundaries.

As we see, the H.263 coding standard provides a number of modes where error resilience is an important consideration. They are designed to facilitate the resilience of the bitstream by both structural and algorithmic techniques.

2.5 MPEG-2

The MPEG-2 standard grew out of the MPEG-1 standard for a wide variety of applications. It is the most widely recognized video coding standard in industry and has seen a number of practical application added to its uses such as HDTV and Digital Satellite System broadcasting. Similar to the ITU-T H Series Recommendations, MPEG-2 defines not only the standard for source coding but also the protocols for the systems level. However certain noticeable differences that impact the error resilience of the streams are apparent in the MPEG-2 System Layer.

Figure 2: Resynchronization markers for a QCIF frame. The GOB headers are shown in white along the left edge of the image while the Slice Structure Mode headers are dispersed throughout the frame at 512 bit intervals in the bitstream.

2.5.1 System Layer

The MPEG-2 systems layer has taken two approaches to the packetization of the data stream: the Program Stream and the Transport Stream approaches. The Program Stream operates close to that of a normal multiplexer where the audio and video data packets are multiplexed with the absolute timestamps being placed within the packets. This technique is however not very resilient to errors and one can face the same problems as described previously in the general description of errors at the multiplexer layer. The alternative is the Transport Stream multiplexing technique [2]. Here the audio and video data are first packetized using variable length packets. Then each of these packets are segmented into smaller packets 188 bytes long that are then transmitted. It can be readily seen that this technique is primarily for ATM networks and has been shown to greatly improve the resilience [2].

2.5.2 Video Coding Layer

MPEG-2 has defined a number of error resilience tools in the video coding layer. The more prevalent ones are an improved Slide Structured Mode, motion vectors for Intra macroblocks, data partitioning, and scalability [6].

Improved Slice Structured Mode - The slice structured mode for MPEG-2 is much more flexible than that defined in the H.261, H.263, and MPEG-1 coding standards. Rather than confining the slice boundaries to be at the beginning of a macroblock, MPEG-2 allows the slice to start at the start of a Transport Packet. This allows for an even finer control of the slice placement and at shorter intervals if needed.

Intra Motion Vectors - An Intra macroblock by default was designed to not convey motion information since none was required in an error-free environment. However since this is not the case in a majority of situations, MPEG-2 allows for motion vectors to be transmitted along with Intra macroblocks. This allows for concealment efforts to be made.

Data Partitioning - The bits in an MPEG bitstream are not all equivalent. For instance, the Picture Start Code, GOB headers, motion vectors, and Macroblock information bits are all much more important pieces of information than the DCT coefficients. This is because an error in one of these information bits will have a much more severe impact on the decoded quality than bit errors in the received DCT information. Because of this unequal data importance within a bitstream, a natural technique of error resilience is what is known as data partitioning. With it the bitstream is separated into two layers of importance allowing each to be protected according to its level of importance.

Scalability - The use of scalability for error resilience is similar to that defined for H.263.

2.6 MPEG-4

MPEG-4 is the first truly multimedia standard designed to consider text, graphics, and video (real and synthetic). The video coding aspect of MPEG-4 are based on the hybrid MC-DCT techniques but are in some ways fundamentally different from the other standards [7]. This is also true of the error resilience capabilities of the MPEG-4 standard where many of the MPEG-2 resilience tools remain along with some newer ones. The error resilience characteristics of MPEG-4 can be divided into three broad categories - resynchronization, data recovery and containment, and concealment.

- Resynchronization - The reasons and basic methods for resynchronization have been stated in the previous sections and remain the same in MPEG-4. The use of a Slice Structured Mode remains from MPEG-2 and is again based on the number of bits rather than the absolute macroblock address. Another approach to resynchronization is the Fixed Interval Synchronization that requires start and resync codes to appear only at valid fixed interval locations within the bitstream.

- Data Recovery and Containment - Data recovery has been greatly facilitated by the use of Reversible Variable Length Codes (RVLC) [7]. These are VLC that can be decoded either backwards or forwards. The use of these codes can help to recover some contaminated data. The use of resynchronization information dispersed throughout the bitstream is an important containment technique.

- Concealment techniques are more a function of the decoder than that of the standard. How a particular decoder chooses to conceal lost or corrupted information can be different from one decoder to another. The more robust the decoder the better it should be able to handle errors gracefully.

Errors in entire macroblocks can be concealed by replacing them with the same macroblock from the previously decoded frame. This is an effective technique if no information about the current macroblock is present. However, if only the texture information is lost or corrupted, then the motion information can be used to conceal the lost texture using a "motion compensated" concealment. The use of motion compensated concealment has been shown to provide approximately 1 dB of gain over simple concealment using the previously decoded frame [2]. We note that data partitioning is important in this case since it will attempt to ensure that the motion information is well protected.

3 THE ERROR-RESILIENT ENTROPY CODE (EREC)

Video coders encode the video data using Variable Length Codes (VLC). Thus, in an error prone environment, any error would propagate throughout the bit stream unless we provide a means of resynchronization. The traditional way of providing resynchronization is to insert special synchronization code words into the bit stream. These code words should have a length that exceeds the maximum VLC code length and also be robust to errors. Thus, a synchronization code should be recognized even in the presence of errors. The Error-Resilient Entropy Code (EREC) [8] is an alternative way of providing synchronization. It works by rearranging variable length blocks into fixed length slots of data prior to transmission. It should be pointed out that the EREC algorithm is not currently a part of any standard.

The EREC is applicable to coding schemes where the input signal is split into blocks and these blocks are coded using variable-length codes. For example, these blocks can be the macroblocks in H.263. Thus, the output of the coding scheme is variable-length blocks of data. Each variable-length block must be a prefix code. This means that in the presence of errors the block can be decoded without reference to previous or future blocks. The decoder should also be able to know when it has finished decoding a block.

The EREC frame structure consists of N slots of length s_i bits. Thus, the total length of the frame is $T = \sum_{i=1}^{N} s_i$ bits. It is assumed that the values of T, N and s_i are known to both the encoder and the decoder. Thus, the N slots of data can be transmitted

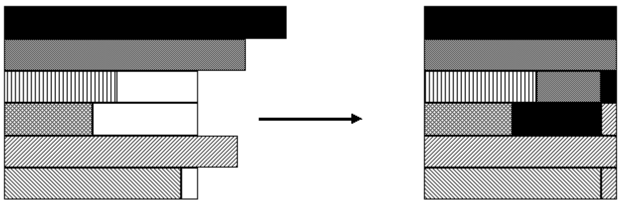

Figure 3: An example of the EREC algorithm.

sequentially without risk of loss of synchronization.

EREC reorganizes the bits of each block into the EREC slots. The decoding can be performed by relying on the ability to determine the end of each variable-length block. Fig. 3 shows an example of the operation of the EREC algorithm. There are six blocks of lengths 11, 9, 4, 3, 9, 6 and six equal length slots with $s_i = 7$ bits.

In the first stage of the algorithm, each block of data is allocated to a corresponding EREC slot. Starting from the beginning of each variable-length block, as many bits as possible are placed into the corresponding slot. In the following stages of the algorithm, each block with data yet to be coded searches for slots with space remaining. At stage n, block i searches slot $i + \phi_n$ (modulo N), where ϕ_n is a predefined offset sequence. If there is space available in the slot searched, all or as many bits as possible are placed into that slot. Clearly, if there is enough space in the slots the reallocation of the bits will be completed within N stages of the algorithm. Fig. 3 shows the final result of the EREC algorithm.

In the absence of errors, the decoder starts decoding each slot. If it finds the block end before the slot end, it knows that the rest of the bits in that slot belong to other blocks. If the slot ends before the end of the block is found, the decoder has to look for the rest of the bits in another slot. Where to look for is clear since the offset sequence ϕ_n is known to the decoder. Since s_i is know to the decoder, it knows the location of the beginning of each slot. Thus, in case one slot is corrupted, the location of the beginning of the rest of the slots is still known and the decoding of them can be attempted. It has been shown that the error propagation is quite low when using the EREC algorithm.

4 ERROR CONCEALMENT

When transmitting a video signal residual errors are inevitable, regardless of the error resilience and channel coding methods used. Thus, ways of mitigating these errors have to be devised. This is the task of error concealment. A number of approaches have been proposed in the literature towards error concealment (see [9, 10] and references therein). Such approaches can be classified into spatial and temporal domain approaches as described next. It is assumed that a motion compensated video coder is used.

4.1 Spatial Error Concealment

Such recovery approaches apply primarily to intra frames or macroblocks, and no temporal information is used (they can also be applied to still images). Information from the neighboring blocks, as well as, prior information about the image is used to "fill-in", or interpolate, or recover the missing information (macroblock, in most cases).

In [11], a technique for recovering lost DCT coefficients is proposed. The missing coefficients are estimated by applying a maximal smoothness constraint at the border pixel of the missing block. First and second order derivatives are used for quantifying smoothness. In [12] this technique is extended for estimating the lost DCT coefficients in inter frames. A similar technique applied in the pixel domain is proposed in [13]. Since the DCT transform is linear, the computation can also be performed in the pixel domain. In [14], the interpolation of the missing pixel values from four corner pixels is proposed. Similar techniques are proposed in [10, 9] for interpolating missing information using a weighted combination of neighboring information, while an iterative adaptive interpolation algorithm is proposed in [15]. A MAP estimation of the missing information is also proposed in [10]. A recursive filter for error concealment is proposed and compared to the MAP estimator in [16]. In [17, 15] an iterative regularized error concealment algorithm is proposed using an oriented high pass operator.

4.2 Temporal Error Concealment

A number of publications have appeared in the literature on this topic. If the motion vectors are adequately protected so that they are received with no errors, the reconstructed or concealed information from the previous frame is motion compensated and serves to conceal lost information in the current frame. Missing DCT coefficients of the displaced frame difference can also be recovered using any of the spatial error concealment approaches. A more challenging situation arises when the motion vectors are lost. Several motion recovery algorithms have appeared in the literature. In a fashion similar to the spatial intensity error concealment algorithms, missing motion vectors are estimated using neighboring motion information. The average [10, 18, 19, 20] and the median [18, 20, 21] have been used as estimates of the missing motion vectors. A MAP motion recovery approach is utilized in [10]. A side matching criterion is used in [22, 18] to estimate the lost vectors. Overlapped motion compensation is then followed in [18], using the estimated motion vector. The motion vectors of the previous picture can also be considered. In the next section a side matching algorithm is outlined [15].

Motion compensated temporal error concealment techniques might provide better results. The combination of spatial and temporal error concealment techniques is described in [9]. Worth mentioning is also that

MPEG-2 provides the capability of temporal error concealment for I-pictures, since the transmission of additional error concealment motion vectors is allowed in MPEG-2.

4.3 A Motion Vector Estimation Algorithm[15]

Inter frames are reconstructed using the motion vectors and the DCT coefficients of the prediction error. Therefore, the loss of the motion vectors seriously degrades the decoded image. This degradation propagates to the subsequent inter frames until an intra frame is encountered. The reconstruction of the lth inter frame takes the form

$$\hat{x}(i,j;l) = \hat{x}(i+Vx, j+Vy; l-1) + \hat{x}_p(i,j;l), \qquad (1)$$

where (Vx, Vy) represents the motion vector for the (i,j)th pixel and $\hat{x}_p(i,j;l)$ denotes the prediction error.

For H.263 in Eq. (1), Vx and Vy are determined by

$$\begin{aligned} Vx &= Dx + Px, \qquad (2) \\ Vy &= Dy + Py, \end{aligned}$$

where Px and Py are the median values of the three neighboring macro blocks, and Dx and Dy are the transmitted differential vectors. Thus, in the case of H.263, the loss of a macro block motion vector propagates to the remaining macro blocks in the frame. In other standards including H.261, the previous motion vector is used for the encoding rather than the median of the neighboring vectors.

With the approach in [15] the motion vector is re-estimated without requiring any differential information from the neighboring blocks. Fig. 4 depicts the idea behind the algorithm. Two frames are shown in it, the current lth frame on the right and the previous $(l-1)$st frame on the left. The gray region represents a missing block in the frame, and the band above and to the left of it the neighboring pixels to be used for the estimation of the lost motion vectors. Such a band is used since only blocks to the left and above the current block have been decoded. The motion vector for the lost macro block is determined as

$$(Vx, Vy) = arg\{\min_{(m,n)\in S_{mv}} \sum_i \sum_j |\hat{x}(i,j;l) - \hat{x}(i-m, j-n; l-1)|\}, \qquad (3)$$

where (i,j) represents the pixels inside the band, S_{mv} denotes the search region in the previous frame, and $|\cdot|$ denotes the absolute value.

Since region matching assumes that the displacement within the region is homogeneous, the support of this region (width of the band) is critical. On the basis of our experiments, 4-8 rows and columns of neighboring pixels result in good matching results.

The algorithm described above was tested under various network conditions. As an example a distorted frame is shown in Fig. 5. Three concealed versions of it are shown in Figs. 6, 7, 8, using the mean and median of the neighboring vectors and the algorithm described above, respectively.

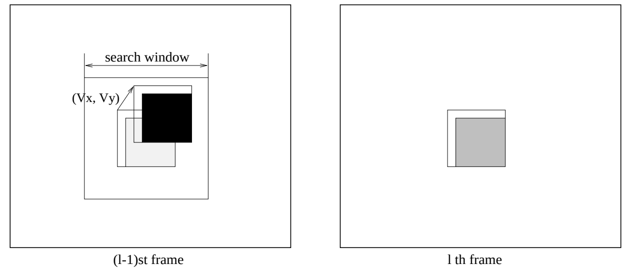

Figure 4: Motion vector estimation.

Figure 5: Missing macro block location of 14th frame of "Foreman" sequence (inter frame)

References

[1] Guido M. Schuster and Aggelos K. Katsaggelos. *Rate-Distortion Based Video Compression, Optimal Video frame compression and Object boundary encoding.* Kluwer Academic Press, 1997.

[2] M. Banham and J. Brailean. *Video Coding Standards: Error Resilience and Concealment*, chapter 5 in Recovery Techniques for Image and Video Compression, A. K. Katsaggelos and N. P. Galatsanos ed. Kluwer, 1998.

[3] International Telecommunication Union Recommendation H.324. Terminal for low bit rate multimedia communication, March 1996.

[4] International Telecommunication Union Recommendation H.223. Multiplexing protocol for low bitrate multimedia communication, March 1996.

[5] International Telecommunication Union Recommendation H.263. Video coding for low bitrate communications, January 1998.

[6] ISO/IEC 13818-2 MPEG II. Information technology - generic coding of moving pictures and associated audio information: video, 1995.

Figure 6: Concealed 14th frame; average motion vector of neighbors, PSNR = 27.13 dB

Figure 7: Concealed 14th frame; median motion vector of neighbors, PSNR = 28.24 dB

Figure 8: Concealed 14th frame; algorithm in [15], PSNR = 29.10

[7] ISO/IEC 14496-2 MPEG IV. Information technology-coding of audio-visual objects: Visual, October. 1997.

[8] D. W. Redmill and N. G. Kingsbury. The EREC: An error-resilient technique for coding variable-length blocks of data. *IEEE Trans. on Image Processing*, 5(4), April 1996.

[9] S. Aign. *Error Concealment for MPEG-2*, chapter 8 in Recovery Techniques for Image and Video Compression, A. K. Katsaggelos and N. P. Galatsanos ed. Kluwer, 1998.

[10] P. Salama, N. B. Shroff, and E. J. Delp. *Error Concealment in Encoded Video Streams*, chapter 7 in Recovery Techniques for Image and Video Compression, A. K. Katsaggelos and N. P. Galatsanos ed. Kluwer, 1998.

[11] Y. Wang, Q.-F. Zhu, and L. Shaw. Maximally smooth image recovery in transform coding. *Trans. on Communications*, 41(10):1544–1551, October 1993.

[12] Q. Zhu, Y. Wang, and L. Shaw. Coding and cell-loss recovery in DCT-based packet video. *IEEE Trans. on Circuits and Systems for Video Technology*, 3(3):248–258, June 1993.

[13] S. S. Hemami and H. Y. Meng. Spatial and temporal video reconstruction for non-layered transmission. In *Fifth International Workshop on Packet Video Visicom'93*, March 1993.

[14] M. Ghanbari and V. Seferidis. Cell-loss concealment in ATM video codecs. *IEEE Trans. on Circuits and Systems for Video Technology*, 3(3):238–247, June 1993.

[15] M.-C. Hong, H. Schwab, L. P. Kondi, and A. K. Katsaggelos. Error Concealment Algorithm for Compressed Video. *to appear: Signal Processing : Image Communication*, 1998.

[16] H. R. Rabiee, H. Radha, and R. L. Kashyap. Error concealment of still images and video streams with multi-directional recursive nonlinear filters. *IEEE Proc. of Int. Conf. Image Processing*, 2:37–40, September 1996.

[17] M.-C. Hong and A. K. Katsaggelos. An iterative regularized error concealment algorithm. *Proc. SPIE Conf. on Digital Compression Technologies and Systems for Video Communications*, pages 82–91, October 1996.

[18] M. J. Chen, L.-G Chen, and R.-M. Weng. Error concealment of lost motion vectors with overlapped motion compensation. *IEEE Trans. on Circuits and Systems for Video Technology*, 7(3):560–564, June 1997.

[19] M. Wada. Selective recovery of video packet loss using error concealment. *IEEE Journal of selective areas in communications*, 7(5):807–814, June 1988.

[20] A. Narula and J. S. Lim. Error concealment for an all-digital high-definition television system. In *SPIE'93*, volume 2094, pages 304–315, 1993.

[21] P. Haskell and D. Messerschmitt. Resynchronization of motion compensated video affected by ATM cell loss. *IEEE Proc. of Int. Conf. Acoust, Speech, Signal Processing*, pages 545–548, March 1992.

[22] W. Lam, A. R. Reibman, and B. Liu. Recovery of lost or errorneously received motion vectors. *IEEE Proc. of Int. Conf. Acoust, Speech, Signal Processing*, 5:417–420, 1993.