Error Correction and Concealment of Block Based, Motion-Compensated Temporal Prediction, Transform Coded Video

A Thesis
Presented to
The Academic Faculty

by

David L. Robie

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Electrical and Computer Engineering
Georgia Institute of Technology
February 2005

Error Correction and Concealment of Block Based, Motion-Compensated Temporal Prediction, Transform Coded Video

Approved by:

Dr. Russell Mersereau, Chair
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Yucel Altunbasak
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Faramarz Fekri
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: March 16, 2005

To my parents,

Leo and Norma Robie,

who continue to offer excellent guidance and direction in every stage of my life.

ACKNOWLEDGEMENTS

A dissertation is not only a body of scientific work, but the chronicles of a trip taken over many years and not without a few divergences. However, along this scabrous path, I have had a prodigious guide and mentor. Dr. Russell Mersereau has, with unending patience, provided wisdom, support and, most significantly, encouragement. During my studies, I was often asked "Who is your advisor" and with great pride I would answer. The responses were always positive such as "He's a great advisor" or "You are lucky", but one response will stay with me always. At a conference half way around the world I was asked the familiar question and I repeated my response to which the French fellow shouted in incredulous disbelief "YOU WORK WITH DR. MERSEREAU". I have received much more than I have given. Thank you.

In addition to my advisor, I would like to thank Dr. Altunbasak or his constructive critiques of my work and his willingness to discuss my problems and offer methods and solutions. He is a indespensible source of knowledge for all his students.

Although, not directly related to my research, many of the other professors at the Center have offered help and encouragement. These include Dr. Faramarz Feckri, Dr. Mark Clements, Dr. Williams, and Dr. Hayes.

I would also like to thank Dr. Scott Wills who willingly donated his time and efforts to the preparation of my Qualifying Exams and Proposal. His outside view of my research was an important element.

Finally, I would like to thank my wife Diane for her help and encouragement along this long arduous trip. Without indepth knowledge of the subject matter, she was always willing to proofread my work, critique my presentations and often accompany me to conferences. Without her help and patience and encouragement, I would have never crossed the finish line.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The use of the Internet and wireless networks to bring multimedia to the consumer continues to expand. The transmission of these products is always subject to corruption due to errors such as bit errors or lost and ill-timed packets; however, in many cases, such as real time video transmission, retransmission request (ARQ) is not practical. Therefore receivers must be capable of recovering from corrupted data. Errors can be mitigated using forward error correction in the encoder or error concealment techniques in the decoder. This thesis investigates the use of forward error correction (FEC) techniques in the encoder and error concealment in the decoder in block-based, motion-compensated, temporal prediction, transform codecs. It will show improvement over standard FEC applications and improvements in error concealment relative to the the Motion Picture Experts Group (MPEG) standard. To this end, this dissertation will describe the following contributions and proofs-of-concept in the area of error concealment and correction in block-based video transmission:

1. A temporal error concealment algorithm using motion compensated macroblocks from previous frames. This research evaluates alternative error measures and demonstrates improved performance with a moderate increase in the computational power, or comparable performance using less computational resources.

2. A spatial error concealment algorithm using the Hough transform to detect edges in both foreground and background colors and using directional interpolation or directional filtering to provide improved edge reproduction.

3. A codec which uses data hiding to transmit three types of error correction information: the length of each macroblock to include byte stuffing values, the final value of differentially encoded DCT-DC values in I frames and the final value of differentially encoded

motion vectors in P and B frames. This information allows the decoder to resynchronize more quickly with fewer errors than traditional resynchronization techniques and also allows for perfect recovery of differentially encoded DCT-DC components and motion vectors. This provides a much higher quality picture in an error-prone environment while creating almost imperceptible degradation of the picture in an error-free environment.

4. An enhanced codec which builds upon the last by improving the performance of the codec in the error-free environment while maintaining excellent error recovery capabilities. Others [1] [30] [16] have attempted to regain resynchronization before the next start code using a technique called Early Resynchronization. While this technique works fairly well, erroneous start points can lead to artifacts in the recovered video. To reduce these artifacts and improve the performance of Early Resynchronization, this codec embeds error detection information in the DCT coefficients using data hiding techniques. Using this parity information the bitstream is resynchronized with fewer errors than traditional Early Resynchronization decoders. In addition, since the data hiding requirements are much lower, error-free image quality is maintained.

5. A method to allocate Reed-Solomon (R-S) packet-based forward error correction that will decrease distortion (using a PSNR metric) at the receiver compared to standard FEC techniques. This encoder is also capable of analyzing channel conditions and varying the source and FEC bit rates to provide improved performance when compared to fixed allocation encoders.

6. Under the constraints of a constant bit rate, the tradeoff between traditional R-S FEC and alternate forward concealment information (FCI) is evaluated. The inclusion of resynchronization on a per packet basis shows significant improvements with little additional overhead for all cases considered. The inclusion of concealment motion vectors represents a more significant overhead, but also may provide improved quality for the receiver, depending upon the complexity of the video being transmitted. Additional possibilities are discussed.

In the following, each of these contributions is described in detail. More specifically, the next chapter presents an overview of block-based, motion compensated temporal prediction, transform coded video while the following chapter discusses previous work in error correction and concealment using block-based codecs and provides a brief introduction to steganography. The temporal error concealment algorithm is discussed in Chapter 3 and the spatial error concealment scheme is described in Chapter 4. A description of the first codec, the Steganocodec is presented in Chapter 5, while the second codec, the Paritycodec is presented in Chapter 6. The application of FEC techniques is presented in Chapter 7 while the evaluation of alternative forward concealment information (FCI) is contained in Chapter 8. Future work and concluding remarks are contained in Chapter 9.

CHAPTER I

INTRODUCTION

1.1 Introduction

The demand for video and other multimedia presentations continues to expand with the Internet and wireless networks becoming important transmission mediums. To use streaming video in these networks requires the use of a compression algorithm in order to decrease the bandwidth to an affordable value, and block-based, discrete cosine transform (DCT) codecs such as the H.263 and the MPEG-X families are widely used (Block Based Codec). These types of encoders achieve compression through the elimination of temporal, spatial and statistical redundancies with the use of motion compensation, block quantization inside a discrete cosine transform, and Huffman run-length encoding. This compression, while reducing redundancies, creates a bitstream that is much less fault tolerant; a single packet loss can cause a loss of synchronization that will be visible over an entire group of pictures (GOP). As a background for a discussion of these errors, their causes and possible mitigation schemes, this chapter will present the MPEG-2 codec as an example of the generic block-based motion compensated DCT encoders followed by examples of the problems generated by bit errors and packet losses.

1.2 Block Based Coding Overview

To perform the compression, the video, which is comprised of a sequence of images, is first divided into groups of pictures each of which contain N frames (Figure 1). The GOP is the smallest segment of video which is self-contained, i.e. does not rely on any other frames for temporal or spatial information. Each group of pictures begins with an intra-coded frame (I frame), which is encoded without reference to other frames. The remainder of the frames may be temporally predicted. These frames are designated as predicted frames (P frame), which are based on the previous I frame or P frame, or bi-directionally predicted frames (B

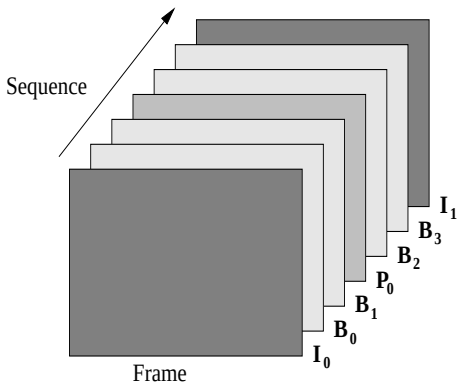


Figure 1: Example of an MPEG-2 Group of Pictures with $N = 6$ and $M = 3$.

frame), which are based on the I frame and or P frame adjacent (in a temporal sense) to the respective frame. In MPEG-2, the determination of P frames and B frames is based on the selection of M, which defines the I-P frame distance. For example, Figure 1 illustrates a GOP with $N = 6$ and $M = 3$. Frames I_0 and I_1 are intra-coded while P_0 is predicted from I_0. In turn, B_0 and B_1 are bi-directionally predicted from I_0 and P_0. Because the B frames are predicted from frames that temporally follow, the frames must be reordered for transmission as shown in Figure 2.

After the frames have been reordered, the individual frames (Figure 3) are coded by dividing the image into slices, or groups of blocks (GOB). A GOB is a group of adjacent macroblocks. Although slices can vary in length to maximize compression they normally begins at the left edge of the frame and extends to the right edge. The macroblock is a 16×16 pixel block that is the basis for intra-coding or motion-compensated prediction. The

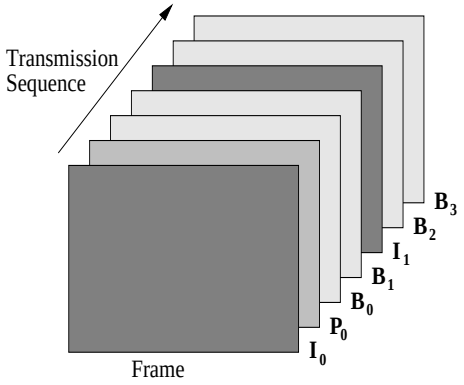


Figure 2: Reordering for frames for transmission in MPEG-2 bitstream.

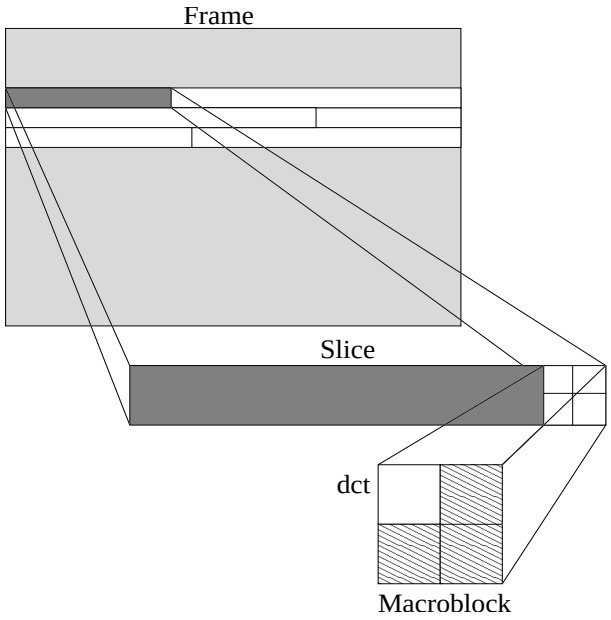


Figure 3: Frames are divided into slices or GOB which, in turn, are divided into 16×16 macroblocks and finally 8×8 transform blocks.

MB is further divided into 8×8 DCT blocks, which are quantized to achieve the desired level of compression. The DCT may be either the transform of actual pixel values (intra) or the residual between the present frame and the temporal estimate (inter or predicted). Macroblocks in I frames are based entirely on intra coding while P frames and B frames may be inter or intra coded. The final step in the compression algorithm is the encoding of DCT values and motion vectors (MV) using a variable length encoder, which in MPEG-2 is a run-length, Huffman coding algorithm. The encoding process is shown in Figure 4.

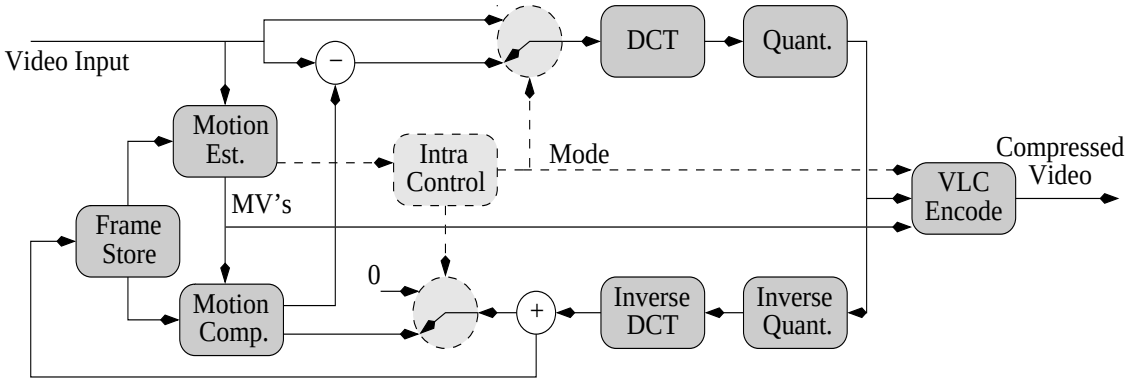


Figure 4: Typical block-based, motion-compensated temporal prediction, transform coder.

1.3 Consequence of Errors in Video Transmission

Errors in video transmission can be propagated both spatially and temporally as illustrated in Figure 5). In this example, the first frame has lost some portion of data, due to a burst error or packet loss, which results in additional spatial loss while the decoder resynchronizes. The error is then propagated temporally to the remaining frames that depend on this frame for prediction information.

Because of temporal and spatial error propagation and the nature of the block-based codecs, it is apparent that all information in the bitstream is not of equal value. Figure 6(a) shows the relative importance of the different components of the video stream. The loss of sequence header information would require retransmission of the sequence header, or for the decoder to wait for the next sequence header causing loss of the entire sequence. Similarity, the loss of a GOP, frame or GOB header may result in the loss of large blocks of information. After the loss of information the decoder must resynchronize and the smallest unit that allows for resynchronization is the slice or GOB; hence a single bit error usually results in a spatial loss of the entire slice as illustrated above. Additionally, the losses in different frames cause varying amounts of temporal propagation. Figure 6(b) presents the relative importance of each frame. Specifically, error in an I frame propagate temporally through the entire group of pictures due to the prediction of the following P frames and B frames (Figure 6(b)). In a similar manner, losses in a P frame propagate errors to all the

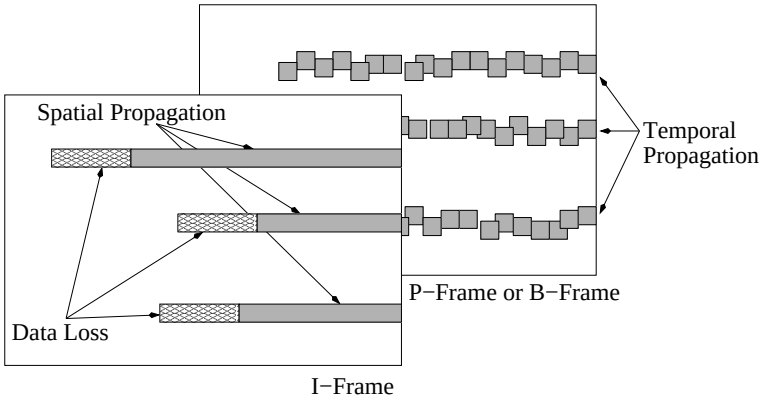


Figure 5: Data losses can propagate spatially through the GOB and temporally through frames in the GOP.

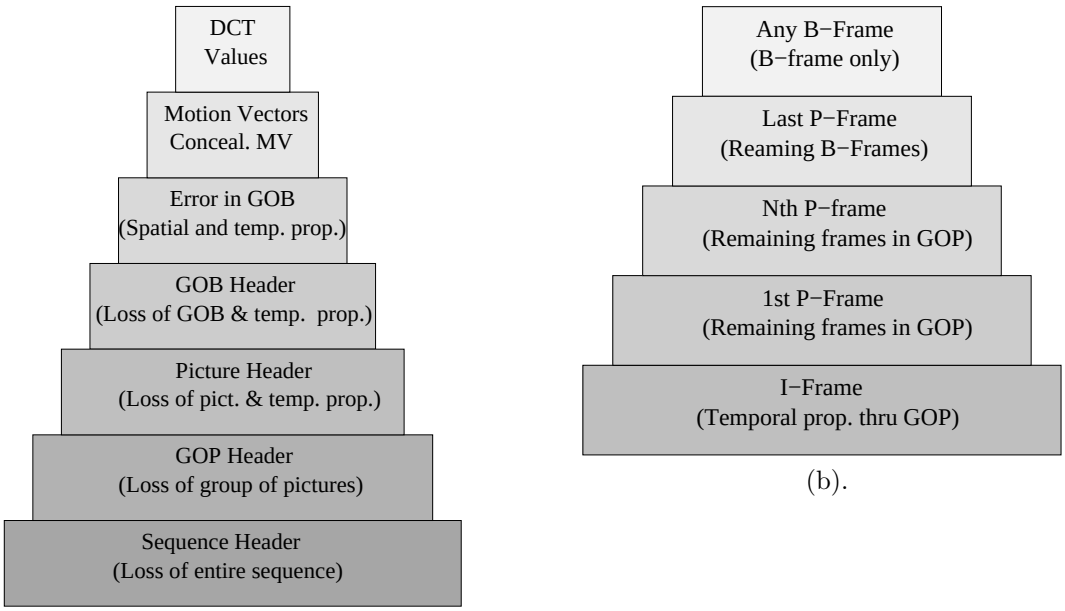


Figure 6: Priority of information in the video bit stream. **(a)** Bits carry different priority information and their loss can have differing effects. **(b)** Losses in different frames will propagate through a differing number of frames.

P frames and B frames following. Fortunately, losses in B frames are confined to a single frame. Thus, the relative value of bits varies greatly in the decoding process.

1.4 Conclusion

In this chapter, the basic structure of a block-based, motion compensated temporal prediction, transform coder has been presented. The relative importance of the data bits was also presented and the effects of errors to include spatial and temporal propagation was demonstrated. The transmission of streaming video must be able to overcome these shortcomings. In the next Chapter a review of research completed in error concealment and correction will be presented. Chapters 3 through 8 will present the contributions of this work and Chapter 9 will present conclusions and areas of future research.

CHAPTER II

PREVIOUS WORK

2.1 Introduction

Block-based, motion-compensated temporal prediction, transform coders such as the H.263 and MPEG-X are widely accepted compression standards capable of compressing video information to an affordable value before transmission. These codecs reduce temporal, spatial and statistical redundancy through motion compensation, block quantization of a discrete cosine transform (DCT), and variable length encoding, and as illustrated in Chapter 1 create a bitstream that is significantly less fault-tolerant than the uncompressed original. For example, single bit errors can cause a loss of synchronization that will be visible over an entire group of pictures (GOP). Hence, when considering the transmission of compressed video over error-prone channels, one must be capable of handling errors and recovering from the resulting loss of synchronization. In this work, the transmission of compressed video over packet-based networks is considered and this review concentrates on packet-based solutions; however, when applicable, research conducted using alternate transmission protocols has been included.

For the transmission of information over packet-based networks such as the Internet, data loss can result from actual packet loss, packet errors or late packets. Actual packet loss typically results from network congestion, while some applications discard packets that contain errors. Additionally, some time-critical applications require packets to be time stamped and discard packets that do not arrive in a specified delay time. Except for these time-critical applications, recovery from packet loss or error is achieved through automatic repeat request (ARQ); however, this option is not always applicable to video transmission. For example in broadcast applications, ARQ may cause network flooding and in real-time transmissions, stringent delay constraints may make retransmission request impractical; therefore, robustness is a key element of any video transmission system.

Increasing the robustness of a video codec can be achieved by increasing the error resilience of the source and channel encoder, improving the decoder's ability to detect errors and conceal their effects and interactive methods between the encoder and decoder. Section 2.2 will discuss error-resilient techniques used by the encoder. Sections 2.3 and 2.4 will discuss error detection and concealment in the decoder followed by a discussion of interactive encoder/decoder error recovery schemes (Section 2.5). Sections 2.6 and 2.7 will present the error-resilient capabilities of the MPEG-2 and MPEG-4 codec which is followed by a brief conclusion.

2.2 Error-Resilient Encoding

In general, the best compression techniques, i.e. those that remove temporal, spatial, and statistical redundancies, make bitstreams vulnerable to channel errors by increasing the dependencies between adjacent frames and pixels. In error-prone environments it may be advantageous to retain some redundancies to improve the error-resilience of the coded bitstream. An example of this in the MPEG-X standards is the inclusion of concealment vectors, which aid in the recovery of intra-coded blocks by providing temporal concealment information computed by the encoder. In a more general sense, almost all error-resilient encoding increases the redundancy of the bitstream, which, in turn, decreases coding efficiency. All practical applications attempt to balance the reduction in efficiency with the increased resilience of the bitstream. Applications capable of responding to time-varying channel conditions are referred to as joint source-channel coding. Before a review of the specific applications of robust encoding and packetization, it is helpful to list a set of criteria for such.

Properties that are advantageous to a robust encoding scheme:

1. Limit the effects of an error to as small a region as possible (limit spatial propagation).
2. Limit the temporal propagation of errors.
3. Provide for a graceful degradation of quality as the bit error rate (BER) or Packet Loss Rate (PLR) increases.

4. Limit the overhead encountered by the error protection and correction scheme.

5. Maintain the best possible image quality.

With these properties in mind, the current state of the art may be divided into the following major areas:

1. Attempt to limit loss due to prediction errors.

2. Attempt to limit the loss of synchronization.

3. Attempt to limit the loss of adjacent blocks through the use of interleaved data.

4. Use of hierarchical coding in conjunction with a prioritized transmission system.

5. Use of forward error correction (FEC).

6. Adding additional information to the transport layer.

In the next sections each of these areas will be discussed with a review of current literature.

2.2.1 Limiting Prediction Errors

One method of improving the error-resilience of the video stream is to limit the use of predicted values. This can be done both spatially and temporally. Spatially, the video is divided into slices or groups of blocks (GOB), which are normally smallest quanta of data with a resynchronization code. Slice headers also initialize the differential encoding (DPCM) values for motion vectors and the DC component of the DCT coefficients. By decreasing the size of the GOB, the amount of information lost due to spatial propagation is limited to the smaller size; however, decreased coding efficiency results from shorter DPCM segments and the additional overhead of GOB headers. In compliance with the MPEG-2 standards, Richardson and Riley improved the error resilience of an MPEG encoder by varying the size of slices in I, P and B-frames [45]. An alternative method, which is not standards compliant, studied an alternate method of grouping macroblocks [36]. In this scheme the authors begin at the center of the picture and denote two GOB's, each moving toward the edge of the picture. Given a uniform distribution of errors, this method increases the

probability that errors will occur closer to the edge of the frame. Limiting temporal error propagation within a group of pictures (GOP) can be achieved by decreasing the size of the GOP; however, the loss of coding efficiency and overhead will yield a higher bit rate for consistent quality. The selection of GOP length is an input to the encoding algorithm.

Another standard compliant method to limit the propagation of prediction errors is the use of additional intra-coded macroblocks. The inclusion of intra-coded MB's limits the propagation of errors both spatially, by reinitializing the differentially encoded MV's and DC values and temporally, by limiting error propagation from a previous frame. The number and insertion location of intra-coded MB's requires information about the present quality of service (QOS) and has been researched by numerous authors [22], [39], [12], and [17].

Multiple threads are another way to restrict the prediction dependency of a bitstream. By partitioning the video data into several independent groups, called threads, and restricting any prediction to within the thread, the bitstream can recover from errors occurring in any individual thread [62]. An example of a two thread system is the encoding the even frames in one thread and the odd frames in another. This is referred to as video redundancy coding.

In general, limiting temporal and spatial predictions decreases error propagation at the cost of decreasing coding efficiency.

2.2.2 Improved Synchronization

Although a start code is the normal resynchronization point in the decoder, the overhead associated with additional start codes can be avoided as demonstrated by Redmill and Kingbury [44] who developed an encoding scheme for use with block-based encoders such as JPEG or MPEG. In this system, the bitstream is divided into N discrete slots of length s_i bits for a total length of $T = \sum_{i=1}^{N} s_i$. Each slot marks the beginning of a block of data. To compensate for the variable length of the block data, excess data from longer blocks ($> s_i$ bits) is appended to the shorter blocks ($< s_i$ bits). Since the data is restructured in a deterministic fashion, it can be replicated by the decoder and the additional overhead is

limited to the transmission of T, N, s_i. A further reduction of the overhead can be realized if all slots can be made equal in length, in which case the only required parameters are T, N which must be sent over an error-free channel (i.e. delivery must be guaranteed). The authors apply this technique to both image and video coding with good results.

Although the use of reversible variable length codes (RVLC) does not improve the resynchronization of the decoder, it does allow for the recovery of a portion of the data that was captured before decoder resynchronization. The MPEG-4 and H-263 standards allow for the use of RVLC for the encoding of motion vectors. Advances in the design of RVLC allow them to encode with near perfect entropy efficiency [61].

Limiting the unusable data that arrives after an error and before the next resynchronization point is such an important issue that later standards such as MPEG-4 allow for the alignment of GOB's with packet boundaries (see Section 2.7). For encoders that do provide this utility, efforts to reduce the loss can also be accomplished at the decoder as will be discussed in Section 2.4.

2.2.3 Data Interleaving

Data interleaving is not supported in most standards, but it can be implemented using multiple threads or hierarchical coding methods. A common type of data interleaving is the interleaving of macroblocks. Concealment schemes used in the decoders depend on spatially proximate information to aid in the recovery of lost MB's. By limiting the loss of adjacent blocks, error concealment schemes, be they spatial or temporal, will have significantly improved performance [69] [54]. As in all applications, the relative value of data interleaving is largely dependent upon the length of the error. In cases where the error burst length is high, data interleaving can mitigate the loss of large sections of contiguous data which present a difficult problems for error-concealment routines.

2.2.4 Hierarchical Coding

In hierarchical coding, video information is partitioned into two or more layers. Three types of scalability are provided in H.263 and MPEG: specifically temporal, spatial or SNR. An additional type of layered coding is data partitioning in which specific data is assigned

to a particular layer in the encoding process. Each of these hierarchical codes and its implementation is presented next.

2.2.4.1 Layered Coding

In layered coding, two or more transport streams are encoded. If the system is prioritized, the base layer which contains the essential information required to provide adequate video quality is given a higher priority. The base layer includes synchronization and decoding data along with motion vectors and the minimum amount of DCT information for the level of quality desired. The lower priority stream(s) will provide enhancements to the base layer and improve quality. The enhancement layers will include enough information for proper decoding and data to improve the temporal, spatial, or SNR quality of the video. In networks that do not provide for prioritized traffic, the probability of loss for the base layer can be decreased with the use of forward error correction tools such as Reed-Solomon FEC. Layered coding is an effective method to provide more graceful degradation of the video and can be accomplished using temporal, spatial or SNR hierarchical coding.

The study in [5] found that the three scalability modes in MPEG-2, namely data partitioning, SNR scalability and spatial scalability, have increasingly better error robustness, in that order, but with increasing overhead as well. Data is also provided indicating the ratio of the bit-rate of the base layer to the total for the different modes and the maximum sustainable packet loss rate.

Wilson and Ghanbari used SNR scalability for a two layer coding system and were able to exploit the inherent inefficiencies in the enhancement layer and adjust DCT coefficients to reduce bit rates to a level comparable with single layer encoding [63].

Hsu et al. [23] demonstrated a spatial scalable coding and error-concealment system for satellite transmission of broadcast TV. In this work the authors combine both spatial and temporal prediction to generate the enhancement residual and propose an error-protection and concealment method. The proposed system claims to provide higher link availability while ensuring uninterrupted TV broadcast.

Layered coding can provide for enhanced reliability and graceful degradation of quality

for transmission over error-prone channels; however, most scalable systems require the base layer be given a higher priority. In situations where this is not possible, Multiple Description Coding can often provide an acceptable alternative.

2.2.4.2 Multiple Description Coding

Another method of layered coding, Multiple Description Coding (MDC), divides information into multiple streams of equal importance. Each coded bitstream, called a description, will provide adequate quality if received; however, each additional description received will improve the quality by some incremental amount. Consider a transmission medium containing more than one parallel path, in which the failure of any individual path is uncorrelated with other paths. This method provides protection against possible channel failure and is also useful for channels with long burst-length errors. MDC requires a large amount of overhead due to the redundancy that exists between the streams, but it may be appropriate for lossy, multiple-path transmissions.

One simple method of implementing MDC is to split adjacent samples among several channels and encoding each of the channels independently; however, more efficient means are available. Vaishampayan [56] created two sub-streams by producing two indexes for each quantization level. For example, by using two quantizers whose decision regions shift by half the quantizer interval with respect to each other. If the quantizer has a bit rate of R bits per sample, the recovery of two streams will have the quantization error of a $R+1$ quantizer, while the recovery of either single stream will provide performance of an R bit quantizer. However, the total bit rate of the $R+1$ quality is $2R$ which can be costly for large values of R. More sophisticated mappings can be designed to improve coding efficiency.

Another method of creating multiple streams, is to use linear transforms that do not completely decorrelate the coefficients. These correlated coefficients can be split between multiple streams, such that any coefficients not received can be estimated from the others. To maintain coding efficiency, coefficients in the same stream should be uncorrelated. This method has been investigated in [58] and [41].

Multiple description coding has numerous advantages; however it is inherently inefficient

because of the redundancy that exists between the steams, and without multiple paths, these advantages are abated. Often, along single channel communications systems, forward error control can significantly reduce the packet loss ratio.

2.2.5 Forward Error Correction

Forward error correction (FEC) has been a very active area of research. In general these studies include, the ability of FEC to correct errors based on different channel models, the allocation of FEC to the video data generated, and the effect of FEC based on differing channel models and conditions. A complete coverage of this topic is not possible, but the more pertinent references are included here.

Biersack [9] defines performance measures and evaluates the effectiveness of FEC over a packet-switched network for three different traffic models in a multicast environment.

Mayer-Patel et al. [35] create a model for the transmission and reception of specific frame types (I,P,B) based on an MPEG encoder. The model is used to evaluate different forward error correction schemes given differing network parameters.

Lee et al. [32] use Reed-Solomon codes to recover lost ATM cells. A four byte RS code is applied to every block of 28 bytes along with data interleaving. Using this method, a loss of up to 4 cells out of 32 can be recovered. A very similar system has been adopted by the Grand Alliance for the broadcast of HDTV.

Ayanogula et al. used FEC on the byte level for error correction for random bit errors and FEC on the ATM cell level for cell loss recovery. The techniques were applied to both single and two layer data with and without cell interleaving. This paper determines that the error correction performed better without cell interleaving.

Cai et al. [4] designed an FEC error control scheme for MPEG-4 transmissions on wireless networks. This unequal error protection (UEP) scheme outperformed equal error protection (EEP) schemes by dividing the data into 2 classes and providing additional protection for the Class 1 information. Cavusoglu et al. [10] develop an Adaptive FEC scheme based on picture type, size of motion vectors and picture rates and compare it to EEP and a static UEP scheme based on frame type.

Kim, Mersereau and Altunbasak [27] develop a system of forward error protection for Set Partitioning in Hierarchical Trees (SPIHT) compression of still images. Mohr, Riskin and Ladner [38] presented a contrasting solution for the same problem.

Zheng and Boyce [67] use an improved User Datagram Protocol (UDP), Complete UDP (CUDP), to improve the transmission of MPEG video. The proposed scheme uses FEC to assist with packet loss with a model that uses packets that contain errors (vs. discarding) in the video decoding process.

Magal, Ianconescu and Meron [36] use unequal error protection by defining three data categories. The first is protected with error correcting codes (ECC), while in the second errors are detected by a cyclic redundancy checks (CRC), and the third has no correction or detection. They also present a method for determining if a start code has occurred even if it has been corrupted by errors.

While transmitting information in a stationary channel, optimal FEC allocation can be determined once and used consistently until a perturbation changes the channel conditions; however, in time-varying systems, optimizing the amount of FEC to the channel at a given time may provide better results. This is an area of investigation referred to as joint source-channel coding.

2.2.6 Joint Source-Channel Coding

Classical Shannon information theory [48] states that the design of the source coder and the channel coder can be separated and the information can be transmitted error free as long as the source information can be represented at a rate below the channel capacity. The channel coder will then add FEC to the compressed stream to enable correction of transmission errors. However, such error-free transmission can only be achieved with infinite delays in implementing FEC, which makes these systems impractical. Hence, joint source-channel coding is an active area of research. Several examples of joint source-channel coding that are relevant to this work are included here. For a more complete review see [60].

Frossard and Verscheure [17] develop a model for pixel-loss probability for a given error and insert additional resynchronization points (slice headers) and intra-coded MB's to

minimize recovery error based on a prediction of recovery. In additional research [18], these authors develop a rate-distortion model to include packet loss and allocate FEC based on the minimum end-to-end expected distortion. In work that uses a distortion model that is similar to Frossard and Verscheure's, Huang and Liang [24] use Rate Compatible Punctured Convolution Codes (RCPC) to allow for unequal error protection to I,P,B frames.

Using an MPEG-4 codec, Wu et al. [64] develop a macroblock loss model to minimize the end-to-end distortion based on feedback of the loss parameters at the receiver. Subbalakshmi and Chen [50] model the channel as an additive Markov channel model, and model the source as an entropy-coded Morkov source. The authors use this modeling in an interactive process that maintains states and metrics for each possible decode of the next bit and chooses the "best" decode.

This is not an exhaustive review of joint source-channel coding and its application to block-based codecs, but does give a review of the papers that most closely align with the work presented here.

2.2.7 Transport Layer Information

After encoding, a video stream must be transmitted. In broadcast or wireless applications the data may be sent directly to the channel modulator, or in packetized systems such as Internet protocol (IP), the video data will be divided into packets with additional information such as start codes and sequence numbers for each packet. The transport layer affords the opportunity for additional error protection. Most of these methods have already been discussed in other sections, but will be reviewed here for completeness.

One method to lessen the effect of burst errors or packet loss is to interleave data as discussed above. Another method is to allow the packets to provide additional information, such as where the next macroblock begins as discussed in Section 2.2.2. It can also be used for error detection with the addition of FEC or CRC as discussed in Section 2.2.5. Finally, all information in the video stream is not of equal importance. Header and start codes are required to begin the decoding process, in DPCM schemes the initial value will affect all other values in the differential coding, and low frequency components are often

more important than high frequency components. At the transport level, these important bits can be protected with additional error correction by using higher priority packets as in layered coding, or transmitted more than once to ensure delivery. These methods are discussed in Sections 2.2.4 and 2.2.1

2.3 Error Detection

After transmission, error detection on the decoder side can be accomplished by either the transport layer or the application layer. Each of these will be discussed in order.

During encoding and packetization additional error detection may be provided by the use of parity bits or CRC codes which allow the transport layer to detect errors to some finite accuracy. If an error is detected, the transport layer can advise the application layer which in turn decides whether to discard or decode the packet. Most applications of Internet and mobile transmission discard packets that contain errors. An example of this in IP networks is RTP which provides real-time, packet lossy, bit-error-free transport [10].

Bitstream errors can also be detected by the decoder. Typically errors appear as violations of syntax such as invalid Huffman decodes, out of range DCT values, more than 64 coefficients in a DCT, or an invalid macroblock address [60]. Other error detection methods attempt to determine if the decoded values are indicative of a natural image. Several heuristic spatial-domain techniques have been developed, but all revolve around comparing the adjacent pixels of consecutive lines or blocks to estimate a smoothness criterion. If values exceed a predetermined threshold, the decoded block is considered to be in error [47][40]. Similar techniques are available in the frequency domain by comparing the values of DCT coefficients with those of adjacent blocks[37].

Be it in either the transport layer or the decoder, after an error is detected, error concealment must be accomplished by the decoder. Techniques to accomplish the error concealment and mitigation are discussed next.

2.4 Error Concealment

Error concealment in block-based decoders has been approached from three primary directions. The first limits the loss of data using a technique called Early Resynchronization (ER) in which synchronization of the data stream is regained before the next start code [1] [30] [16]. The second is temporal concealment, which attempts to use the data from past frames to fill in lost blocks. Simple temporal concealment schemes copy the missing macroblocks (MB) from a previous frame, while more advanced systems use motion compensation to improve performance [2]. The third technique is spatial concealment, which uses surrounding pixels to reconstruct lost data. Neither resynchronization nor temporal concealment can conceal all errors satisfactorily because resynchronization always results in the loss of at least one macroblock and temporal concealment fails in occluded regions and during scene changes. Therefore, an effective spatial concealment algorithm is required in any complete error concealment scheme. Spatial concealment based on interpolation works well in flat regions [2], but fails in textured regions or at edges. Directional interpolation and directional filtering perform much better in these areas [28] [46]. An excellent review of error concealment can be found in [60].

2.4.1 Early Resynchronization

In previous research, the author determined that a significant component of error recovery was the decoder's loss of synchronization. If the decoder can be rapidly resynchronized, the propagation of errors is significantly reduced and a technique to accomplish this is Early Resynchronization.

Early Resynchronization [30] [16] [1] [31] is an attempt to recover information that has been received by the decoder, but is before the next resynchronization point in the bitstream (Figure 7). To begin, all bits between the error and the next resynchronization point (e.g. a start code in MPEG-2) are stored. Using the first bit as a starting point, the decoder attempts to decode the string. If it cannot decode successfully, it discards the first bit and attempts to decode again; if the decode is successful, it assumes that it is valid and uses the decoded information for error concealment. It continues until the string is decoded

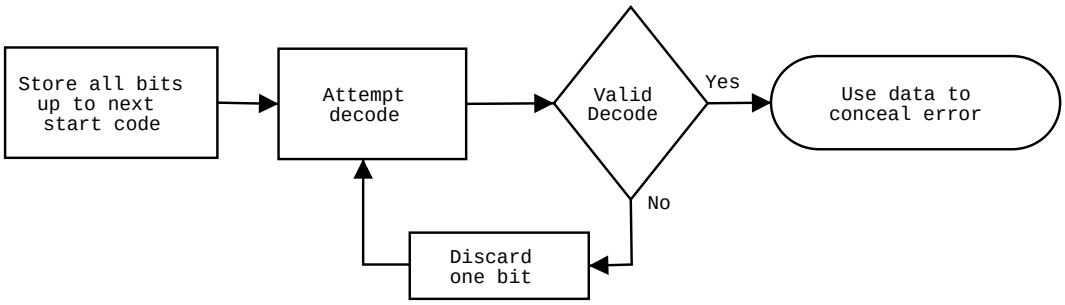


Figure 7: Early Resynchronization (ER) algorithm.

without error or all bits are discarded. Bitstream syntax errors such as invalid Huffman decodes or syntax violations of the bitstream such as out of range DCT values, more than 64 coefficients in a DCT, or an invalid macroblock address increment are used to indicate an invalid decode. In this work, the term valid means that no syntax errors were generated by the decoder, whereas error-free means not only correct syntax, but correct decoding of all values associated with the string.

In practice, as the string is decoded, the probability of decoding an error-free macroblock increases as more bits are decoded; end of block indicators in the DCT's and bit stuffing at the end of each macroblock assist the decoder in regaining synchronization. Conversely, the first several macroblocks are most likely to be valid decodes, but contain errors. Eliminating or detecting these errors is important to improving the resynchronization process and techniques for accomplishing this are presented in Chapter 6. In the next section the recovery of differentially encoded values is discussed.

2.4.2 Differentially Encoded DCT-DC Values and Motion Vectors

Although regaining synchronization is very important, the loss of differentially encoded DCT-DC values in I-frames can create very noticeable artifacts. Exact recovery of the DC values is not possible and estimating them requires computationally intensive methods that are not always effective. One such method [30] assumes a smooth boundary between the recovered slice and adjacent slices, matches the boundary pixels between macroblocks and selects an initial DCT-DC value that will minimize the mean squared error (MSE) across this boundary.

Similarly, in P-frames and B-frames, the loss of the differentially encoded motion vectors causes very noticeable errors. Recovery methods for motion vectors include [2] using the motion vectors from adjacent slices, or searching a limited range of motion vectors and selecting one based on the mean absolute difference (MAD) or MSE across the boundary pixels of adjacent macroblocks. These methods are rarely able to exactly recover the lost motion vectors.

The recovery of these differentially encoded values is inexact, computationally intensive and often produces noticeable artifacts. The inability to recover these values and the inevitable loss of entire MB's in packet networks necessitates temporal and spatial error concealment schemes.

2.4.3 Temporal Error Concealment

Temporal error concealment uses the data from previous or future frames to conceal the lost data in the recovered frame. The simplest form of temporal error concealment simply copies the data from the last received frame to the current frame. This technique provides acceptable results with minimal camera motion and a static recovered area such as the background in a still-mounted camera video. This technique works poorly in most other cases.

A more advanced technique is to use motion compensation to improve the performance of the temporal concealment. This idea has been investigated by several authors. Aign [1], used the motion vectors (MV) of the eight surrounding macroblocks as candidate motion vectors, and chose the MV that minimized the mean squared error across the boundary pixels between the missing macroblock and its neighbors. Zhang et al. [65] expand upon the neighbor method and use a global search in an attempt to improve the performance of the motion compensation. They also use an alternative error measure that uses the pixels surrounding the missing macroblock to a depth of either two or eight. Tskekeriou and Pitas [55] use search methods similar to Zhang, but use the entire macroblock above, below or both to minimize the mean absolute difference between the pixels and select a motion vector for concealment. Finally, Su and Ho [52] use the optical flow of the region above the

missing macroblocks to estimate the motion for temporal concealment with very modest improvements over the above methods.

Temporal error concealment techniques perform well in areas of no apparent motion or very consistent motion (e.g. panning camera); however, they fail when motion is excessive, the concealed block is occluded in previous frames, and during scene changes. These frailties can often be overcome with a robust spatial concealment scheme.

2.4.4 Spatial Error Concealment

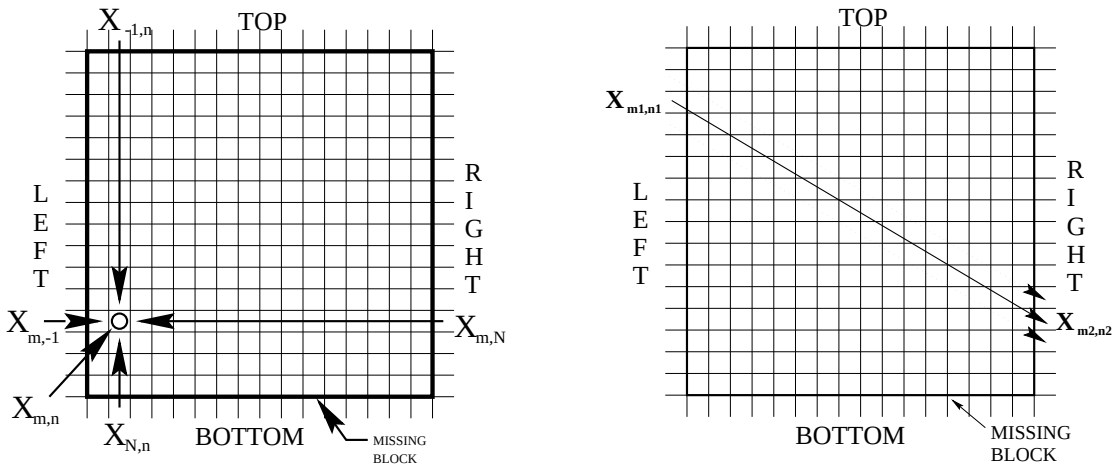
As stated above, spatial error concealment uses the data in the present frame, i.e. adjacent blocks, to attempt to reconstruct the lost data. The simplest spatial scheme is based on bi-directional interpolation of the boundary pixels of the surrounding blocks as shown in Figure 8 and described in Equation 1 where N is the width of the block, λ is $\frac{1}{2}$, and μ_n is the distance operator defined in Equation 2 [2].

$$\begin{aligned} x_{i,j} &= \lambda(\mu_i x_{i,-1} + (1-\mu_i)x_{i,N}) + \\ &+ (1-\lambda)(\mu_j x_{-1,j} + (1-\mu_j)x_{N,j}) \end{aligned} \tag{1}$$

$$\mu_n = 1 - (\frac{1}{2N} + \frac{n}{N}) \quad \text{for } 0 \leq n \leq N-1 \tag{2}$$

However, this type of spatial concealment only works well in areas of little texture with no significant edges. Conversely, due to the low pass nature of interpolation, it works very poorly in areas containing edges or significant texture. The overall appearance is blurred and edge features are smeared.

In an effort to further exploit the correlation between neighboring blocks, others have used directional interpolation to improve the edge features. Directional interpolation is illustrated in Figure 8 where the values along the direction of interpolation are distance weighted, as above, but, only in one direction. The value given to pixels is shown in Equation 3 where N is the length of the line between edge pixels x_{m_1,n_1}, x_{m_2,n_2} and μ_n is



(a) Bi-Directional.

(b) Directional.

Figure 8: Spatial error concealment using interpolation. **(a)** illustrates bi-directional interpolation and **(b)** illustrates directional interpolation.

the distance measure defined in Equation 2.

$$x_{i,j} = (\mu_i x_{m_1,n_1} + (1 - \mu_i) x_{m_2,n_2})$$
$$\forall x_{i,j} \text{ on the line between } x_{m_1,n_1} \text{ and } x_{m_2,n_2} \tag{3}$$

A key element of directional interpolation is selection of the direction. Suh and Ho [51] use the boundary pixels of the missing block to determine one of four possible directions to interpolate. Kwok and Sun [28] used a Sobel gradient operator and a voting scheme over the pixels surrounding the macroblock to choose from eight possible directions. Their directional interpolation scheme was similar to the one above with the exception of using more than the first edge value, but with the same type of distance operator.

Another method of spatial concealment uses projections onto convex sets (POCS). Sun and Kwok [53] use the eight neighbors of the missing block and a Sobel operator to classify a block as smooth or containing an edge. If it has an edge the direction is chosen in one of 8 equally spaced directions in the range $[0^o, 180^o]$. Then the recovered values are forced onto two constraints, either smooth or edge and limited to the range $[0, 255]$. The algorithm normally converges in 5–10 iterations.

Frequency-domain techniques have been used to recover the low frequency values of lost

DCT coefficients. The recovery of high frequency components has not proved successful so the same loss of detail is noticeable in these techniques. Hasan et al. [21] divided the image into 64 one dimensional signals to represent each of the frequency components of the 8×8 DCT to recover the lost components.

Again in the frequency domain, several authors [69] [42] have used a smoothness constraint on the image intensity to recover the low frequency components of the DCT. The high frequency components must be either recovered from the bitstream or set to zero (otherwise the smoothness constraint would zero them).

In a combination of spatial and temporal concealment Zhu et al. [69] used the previous images to improve the estimates of low frequency DCT coefficients again using a smoothness constraint.

In general, all spatial error concealment schemes work well in smooth areas, but to be effective in areas that contain edges, effective directional interpolation algorithm is required and the selection of the correct direction is the most important element. In the Sun and Ho work, four directions are usually not enough and in the Kwok and Sun algorithm, the direction can be incorrectly chosen based on edges that do not intersect the missing macroblock. In this work, an implementation using the Hough transform to isolate edge features and improve the performance of directional filtering schemes is presented in Chapter 4.

Each of the concealment schemes discussed, early resynchronization, temporal or spatial error concealment will have different computational complexity depending on the implementation. While considering the complexity of the algorithms, one must consider the importance of the frame in relation to the entire video. For example, since I-frames will influence the entire GOP, error concealment in these frames is the most important. This is followed by P-frames since they influence the B-frames before and after, and finally, since B-frames are self contained, any type of rudimentary concealment is usually satisfactory.

In the next section, the use of feedback to improve error concealment and recovery will be discussed.

2.5 Error Concealment Using Feedback

As previously stated, during real-time video transmission, attempting to recover damaged packets through a retransmission request is not viable because of the streaming nature of video and the latency of the network. However, if a feedback channel is available, the encoder can react to network conditions and in some cases limit the effects of the channel errors while still maintaining the best possible quality. In this section techniques based on decoder feedback will be reviewed.

In an effort to limit the effects of an error the decoder may request the next frame be intra-coded. This can have a large effect on the bit rate so Wada [57] suggested two techniques for limiting this problem. In each case after the error is detected the decoder notifies the encoder of the location of the error. In the first method, the encoder calculates the location of the damaged blocks in the current frame being encoded and codes those areas without the use of prediction. In the second scheme, the encoder uses the same error concealment scheme as the decoder for the affected areas, and then bases all prediction on the error concealed frame. Both of these schemes limit the propagation of an error in the video sequence.

Another method defined in the H.263 standard allows for the use of multiple reference frames. The encoder and decoder keep a list of which frames have arrived error-free, and the encoder can base its prediction on error-free frames [59].

The use of retransmission of video data is typically considered unrealistic; however, several have investigated the idea. One technique is to freeze the frame while awaiting for the retransmitted data to arrive. Of course, this introduces a fixed delay for the display of all subsequent frames and is termed the accumulation delay.

Another method to recover from an error through retransmission of data, but to avoid the delay, is to first conceal the error at the decoder while requesting retransmission [68] . Then while awaiting the retransmitted data, the decoder tracks the location of the errored pixels. As the data arrives, the decoding is reaccomplished and the pixels (that have been tracked) in the current frame are replaced by the correctly decoded pixels. This again limits the propagation of the error for a period longer than the retransmission time. The author

does note however, that for prediction based on fractional pixel accuracy, the tracking of the pixel errors can be very complex.

One final retransmission method for use in very lossy channels calls for multiple copies of lost packets to be sent, thereby increasing the probability of error free arrival of the packet. The details can be found in [60].

It is important to note that in most retransmission schemes the authors try to maintain a constant bit rate, by decreasing the output of the encoder (i.e. lower quality) to account for the bandwidth used for retransmission. In this way, the network is not flooded by retransmission data as the error rate rises.

The next two sections discuss the error-resilient coding techniques provided in the MPEG-2 and MPEG-4 standards.

2.6 MPEG-2: Error Resilient Coding Techniques

The MPEG-2 standard provides for a vast range of qualities and can encode video ranging from television quality all the way to broadcast production quality. It is the standard that is frequently used in the popular DVD movie and in numerous satellite broadcast systems. Bit rates can range from 1 Megabit/sec. (Mbps) to in excess of 15 Mbps. Although MPEG-2 is a very versatile standard, it contains very few error-resilient tools (ERT).

The only error-resilient tools available in the MPEG-2 standard are variable slice length, concealment motion vectors (MV), and scalability. Each slice begins with a slice start code which provides a resynchronization point for the decoder. At this point the decoder can reinitialize differentially encoded MV's, DC components, quantization levels, and identify the type and location of the next MB to be decoded. Adding additional start-of-slice headers limits the spatial propagation of errors by resynchronizing the decoder. MPEG-2 also allows for the inclusion of concealment MV's for intra-coded blocks to allow for motion compensated error concealment. Typically concealment MV's are transmitted with the MB's of an adjacent slice to facilitate recovery. The last ERT available in MPEG-2 are four types of scalability which include spatial, temporal, SNR, and data partitioning, which allow the video to be reconstructed from two or more data streams. Typically data is divided into

a base layer which provides acceptable quality, and an enhancement layer which improves the picture by adding improved resolution. The use of scalability with differing QOS's can provide for significant error resilience in the bitstream.

Research in correcting the frailties of the MPEG-1 and MPEG-2 bitstream provided the developers of later video codecs with a significant toolbox of error-resilient techniques, which have been included in the MPEG-4 and H-263 standards.

2.7 MPEG-4: Error Resilient Coding Techniques

The most recent video coding standard in the MPEG-X family is MPEG-4 which is designed for audio-visual encoding of multimedia applications and allows for interactivity, high compression and universal accessibility and portability of content. Target bit rates are between 5-64 kilobits/sec. (Kbps) up to 2 Mbps for television applications. The MPEG-4 standard incorporates a set of error resilient-tools that include:

1. Video packet resynchronization;
2. Data Partitioning (DP);
3. Reversible Variable Length Codes (RVLC);
4. Header extension code (HEC);

2.7.1 Video packet resynchronization

The loss of synchronization in the bitstream can lead to significant spatial error propagation as the decoder searches for the next resynchronization point (Sections 2.2.2 and 2.4.1). In a typical block-based video codec, resynchronization points are aligned with macroblock boundaries not with packet boundaries as shown in Figure 9(a); however, MPEG-4 allows the resynchronization points to be aligned with packet headers (Figure 9(b)). The header information provided after the resynchronization code allows the decoder to restart the decoding process if required and includes the macroblock address, and the quantization parameter and a header extension code (HEC) which is discussed below (Figure 10). Additionally, these resynchronization points reset prediction values for motion vectors and DC

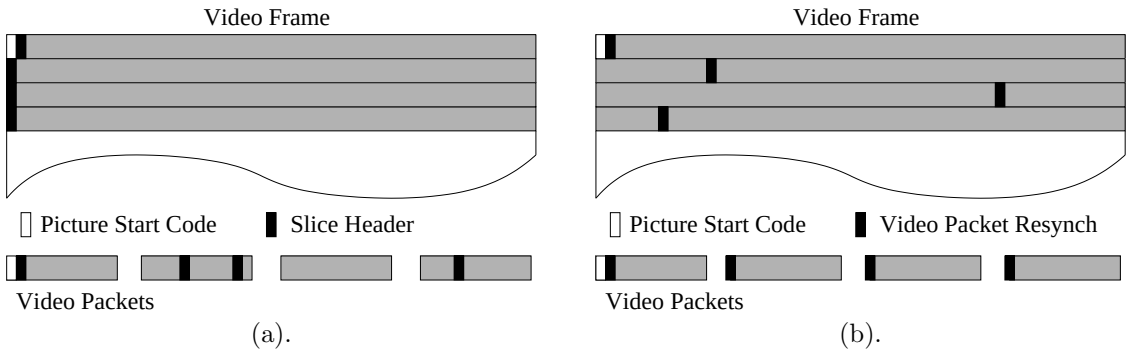


Figure 9: Alignment of resynchronization points relative to packetization. **(a)** Macroblock alignment of start headers. **(b)** Video packet alignment of resynchronization points.

components as necessary.

In addition to MB aligned and packet aligned resynchronization markers, MPEG-4 also allows for fixed interval resynchronization which requires the start of a video packet to appear only at fixed intervals in the bitstream. This technique prevents start code emulation.

2.7.2 Data partitioning (DP)

In typical block-based codecs, macroblock information is coded consecutively with each macroblock providing information such as type (inter or intra), quantization level, motion vectors, and DCT texture information. MPEG-4 provides for data partitioning by transmitting motion vectors for all macroblocks, followed by a 17 bit motion boundary marker (MBM) and texture information for all included MB's. This data partitioning enhances the resilience of the video packet. If an error occurs before the MBM, the decoder can find the MBM and decode the remaining texture information. Similarly, if an error occurs after the MBM, all MV's will be available for error concealment. Using this method, error concealments schemes do not have to recover from the loss of both MV's and DCT information simultaneously.

As discussed above, reversible variable length codes (RVLC), can be decoded in either direction; therefore, after detecting an error, the decoder can buffer the data until the next resynchronization marker, at which point, texture information can be decoded in reverse. This allows for the recovery of the texture information for as many MB's as possible.

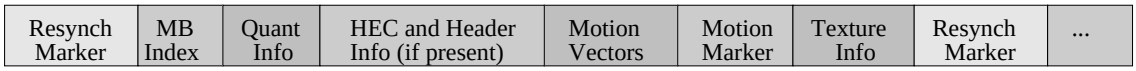


Figure 10: Header extension code (HEC) format.

2.7.3 Header extension code (HEC)

The header extension code is a single bit that informs the decoder that additional video object plane (VOP) information is present. The VOP information includes timing information, temporal reference, VOP prediction type, and other information. The inclusion of this information allows for the decoder to recover a portion of the information of a VOP even in the case of VOP header loss which would normally necessitate discarding the entire VOP packet.

2.7.4 Reversible Variable Length Codes (RVLC)

In conclusion, it can be seen that the error resilient tools of the MPEG-4 codec can provide additional robustness to the video stream; however, in a packet lossy environment, the use of FEC in conjunction with error concealment is still required to provide acceptable video quality.

2.8 Summary

In this Chapter, a review of error correction and concealment with respect to block-based, motion-compensated temporally predicted, transform coding has been presented. This review includes error resilient encoding techniques, methods for joint source-channel coding, the uses of feedback to enhance error recovery and error concealment techniques used in decoders. The inadequacy of all the decoders to provide acceptable video has been the motivation for this work and in the remaining Chapters, the original contributions of this work will be presented.

CHAPTER III

MOTION-COMPENSATED TEMPORAL ERROR CONCEALMENT

3.1 Introduction

Temporal error concealment uses the data from previous or future frames to conceal the lost data in the recovered frame. The simplest form of temporal error concealment simply copies the data from the last received frame to the current frame. This technique provides acceptable results with minimal camera motion and a static recovered area such as the background in teleconference applications; however, it performs poorly in most other cases. More advanced techniques use motion compensation (MC) to improve the performance of the temporal concealment; however, search methods for determining the correct motion compensation and evaluation measures for judging the motion compensation results are an integral part of this technique. The results presented here investigate and compare several possible search methods and error measures.

The next section will present the search techniques and error measures used by others and proposed here. Section 3.3 will present the results and Section 3.4 will present the conclusion.

3.2 MC Search Techniques and Error Measures

As discussed above, simple temporal replacement from the preceding frame is acceptable in only the simplest cases. When motion becomes complex, motion-compensated temporal concealment offers an acceptable alternative. Several different methods have been evaluated. Aign [1], uses the motion vectors (MV) of the eight surrounding macroblocks as candidate motion vectors, and chooses the MV that minimized the mean squared error across the boundary pixels between the missing macroblock and its neighbors (Figure 11). Zhang et al. [65] expand upon the neighbor method and use a global search in an attempt to

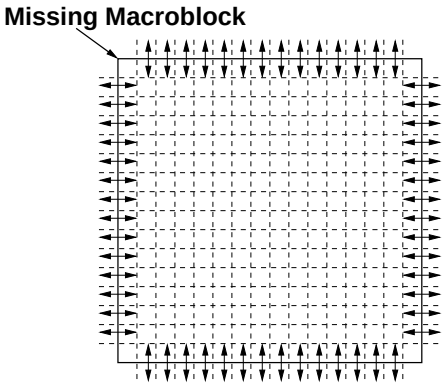
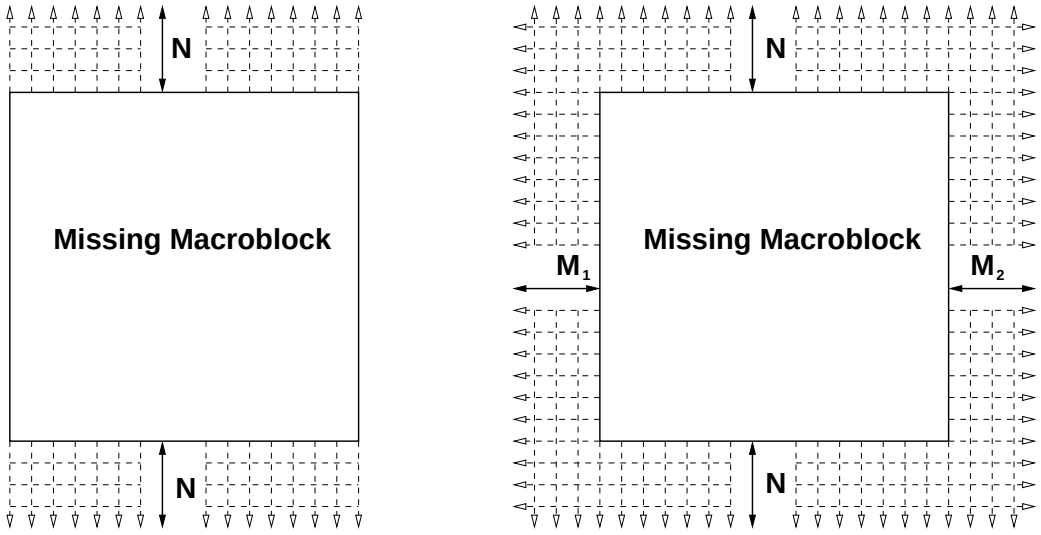


Figure 11: The error measure used by Aign is the MSE across the boundary pixels of the missing macroblock.

improve the performance of the motion compensation. They also use an alternative error measure that includes the pixels surrounding the missing macroblock to a depth of either two or eight. Tskekeriou and Pitas [55] use global search methods similar to Zhang, but use the entire macroblock above, below or both to minimize the mean absolute difference between the pixels and select a motion vector for concealment. Finally, Su and Ho [52] use the optical flow of the region above the missing macroblocks to estimate the motion for temporal concealment with very modest improvements over the above methods.

The error measure defined here is a more flexible and versatile expansion upon the already proposed error measures. This measure is the Mean Absolute Difference (MAD) of the surrounding pixels as shown in Figures 12(a) and (b). In Figure 12(a), the one-dimensional (1-D) block MAD evaluates the mean absolute difference of the N pixels above and below the lost macroblock. The 1-D error measure is the same as Tsekeridou et al. if $N = 16$. In Figure 12(b), the two-dimensional (2-D) block MAD evaluates the mean absolute difference of the N pixels above and below the lost macroblock and the M_1 pixels to the left and the M_2 to the right of the missing MB. The 2-D block MAD is similar to Aign's error measure if $N = M_1 = M_2 = 1$ with the exception that MAD is used instead of MSE. In this research, no consistent or significant difference was noted between MAD versus MSE. The 2-D error measure is the same as Zhang et al. if $N = M_1 = 2, 8$ and $M_2 = 0$. Additionally, the 1-D can be viewed as a special case of the 2-D with $M_1 = M_2 = 0$. In all results presented here $N = M_1 = M_2$ except as noted below.



(a) 1-D Error Measure.

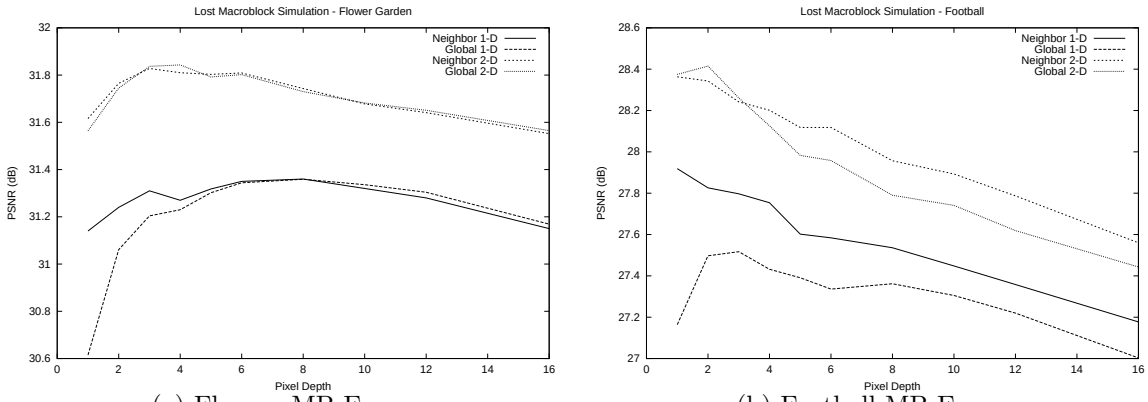
(b) 2-D Error Measure.

Figure 12: MAD error measure for **(a)** one dimensional and **(b)** two dimensional computations.

In this work, the loss of both a single macroblock and the loss of group of blocks (GOB) is simulated using three different sequences, FLOWER GARDEN, TABLE TENNIS and FOOTBALL. The MSE measure of Aign [1] and a boundary matching algorithm similar to Tsekeridou et al. [55] and Zhang et al. [65] are compared for performance and computational intensity. A "Best MC" based on the motion vector generated by the encoder for the missing block is also presented as a benchmark.

3.3 Results

In the macroblock loss scenario, every other macroblock is lost in the odd slices. This simulation is valuable in two cases. The first is the loss of a single macroblock due to a bit error after which the sequence is resynchronized, while the other is the loss of a larger slice of data while interleaving the macroblocks. For each sequence, the 1-D and 2-D error measures are evaluated using both a global search and a search of the motion vectors from the neighboring macroblocks (referred to as neighbor). As shown in Figure 13(a) for the FLOWER GARDEN sequence the 2-D MAD error measure outperforms the 1-D measure by almost $0.6dB$ while the difference between the neighbor and global routines is almost

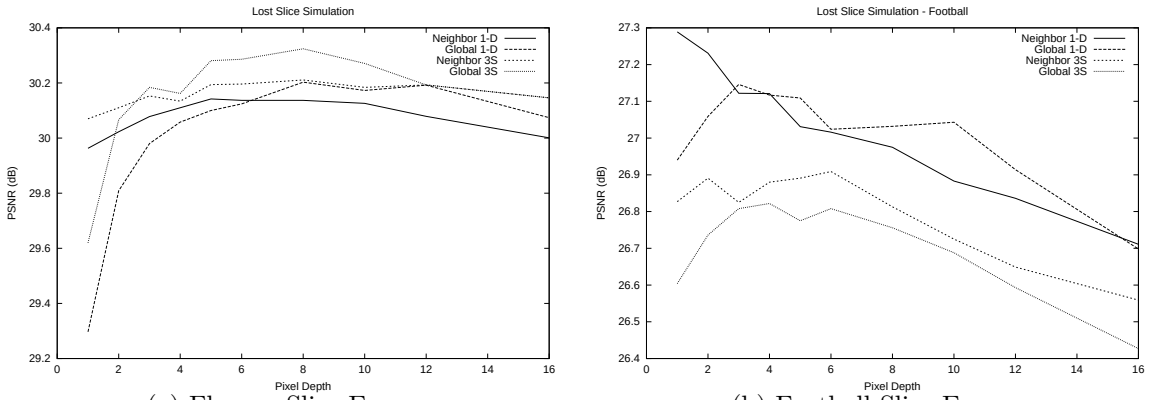


(a) Flowers MB Error. (b) Football MB Error.

Figure 13: Macroblock loss scenario. PSNR as a function of pixel depth for **(a)** FLOWER GARDEN **(b)** FOOTBALL.

insignificant. The similarity of the two search methods is due to the highly correlated motion in this panning sequence; however, as the pixel depth, N and M increase, the performance of the error measures climb to an optimal level and then decrease as the pixel depth continues to increase. Although the decrease is modest in the FLOWER GARDEN sequence, it is more pronounced in the FOOTBALL sequence where the motion is more chaotic (Figure 13(b)). This is the result of the decrease in the correlation of the motion between pixels as the distance between pixels increases in this complex sequence. Hence, limiting N is valuable not only to decreasing the computational load, but may also improve the temporal concealment performance in complex sequences. Results for the TABLE TENNIS sequence were between the results of the FLOWER GARDEN and the FOOTBALL sequence which is expected since it has a combination of simple staring along with complex zooming and player motion. The results presented here are consistent with single MB loss or if the neighbors of the lost MB have been received correctly; however, it is common in video transmission to lose an entire slice or GOB's. The results for these tests are presented next.

In the GOB loss scenario (Figure 14), one half of the slice is lost in each of the odd slices simulating the loss of a number of sequential packets or the inability to resynchronize the bitstream. For each sequence, the 1-D and 2-D error measures are evaluated with the exception that for the 2-D measure, the pixels to the right of the missing macroblock are



(a) Flowers Slice Error. (b) Football Slice Error.

Figure 14: Slice error scenario. PSNR as a function of pixel depth for **(a)** FLOWER GARDEN and **(b)** FOOTBALL.

excluded (i.e. $M_2 = 0$) from the computation since these pixels are normally not present in a GOB loss scenario. This error measure is referred to as 3-Sides (3-S) to delineate from the normal 2-D MAD. In Figure 14(a) the results for the FLOWER GARDEN sequence show that the 3-S outperforms the 1-D error measure just as in the single macroblock loss case, but the global search was able to perform only slightly better than the neighboring vectors. This may be the result of having fewer motion vectors to evaluate (6 neighbors vs. eight due to the lost macroblocks to the left and right). In the FOOTBALL sequence the results are not as easy to discern. The best results where achieved by the 1-D neighbor search with the 1-D global giving the next best performance. Again, there is a decrease in performance as pixel depth increases because of insufficient correlation between pixels as displacement from the missing macroblock increases. As in the macroblock error scenario, limiting the depth of the pixel search decreases the computational load and improves performance in sequences with complex motion.

In Table 1 the results for the macroblock loss scenario are presented using a number of different search methods for the three sequences. In each column, the best results for the type of search are displayed with N, the pixel depth, and the Y PSNR. In the first row, "No MC" represents temporal replacement with no motion compensation. The next row, "Best MC" uses the motion vectors generated by the encoder for the lost macroblock. This

Table 1: Y PSNR for three sequences based on the loss of isolated macroblocks.

	Flowers (N/dB)	Tennis (N/dB)	Football (N/dB)
No MC	0/23.12	0/30.06	0/24.75
Best MC	NA/32.53	NA/36.73	NA/29.97
Aign	1/30.10	1/34.39	1/28.11
1-D Neighbor	8/31.36	6/34.51	1/27.92
1-D Global	8/31.36	5/34.58	3/27.51
2-D Neighbor	3/31.83	2/36.17	1/28.36
2-D Global	4/31.84	2/36.30	2/28.41

method is not possible without additional side information, but provides an upper bound for motion-compensated temporal replacement. The "Aign" column refers to the author and the most common type of motion-compensated temporal concealment (Figure 11). The last four rows show best results of the four search cases previously presented, i.e. 1-D neighbor and global searches and the 2-D neighbor and global searches. Several important issues can be gleaned from the results. First, the proposed error measures are an improvement over the state of the art (Aign) in all cases except the 1-D football sequence. Next, as the complexity of the motion increases the pixel depth of the error measure must decrease to improve results. In the MB loss tests, the 2-D error measure is a significant improvement as expected. Finally, the "Best MC" is a $0.5 - 1.5dB$ improvement over every search method. This result will be useful in future work.

Table 2 is presented for completeness; it contains the results for the GOB loss scenario which are very similar to the results for the macroblock loss presented above. Each column contains, N, the pixel depth and the Y PSNR for the best search. In the rows, "No MC" represents temporal replacement with no motion compensation, "Best MC" uses the encoder-generated motion vectors for the lost macroblock, "Aign" selects the best neighboring motion vector based on MSE across the boundary pixels and the last four rows show best results for the four search cases previously presented, i.e. 1-D neighbor and global searches and the 2-D neighbor and global searches. The conclusion from these tests are first, the proposed error measures are an improvement over the state of the art (Aign) in all cases. Next, the relationship between the complexity of the motion and the pixel depth

Table 2: Y PSNR for three sequences based on slice errors.

	Flowers (N/dB)	Tennis (N/dB)	Football (N/dB)
No MC	0/22.66	0/31.17	0/24.37
Best MC	NA/32.12	NA/36.53	NA/29.43
Aign	1/27.72	1/33.26	1/26.04
1-D Neighbor	5/30.14	2/35.04	1/27.29
1-D Global	8/30.20	5/35.32	3/27.15
3-S Neighbor	8/30/21	4/35.00	6/26.91
3-S Global	5/30.28	3/35.36	4/26.82

again held true, but the improvement shown by increasing or decreasing depth is much less noticeable. This is portrayed by the flatness of the curves in Figure 14(b) where the differences are on the order of $0.01dB$. In contrast to the MB loss scenario, the three side error measure was not always an improvement. This may result from using the macroblock to the left, which is also temporally concealed and hence may be in error. Finally, the "Best MC" is again a significant ($1.2dB - 2.3dB$) improvement over every search method.

3.4 Conclusion

In closing, the error measures proposed here are a significant improvement over the measure proposed by Aign; however, increasing complexity did not always yield improved performance. In sequences with complex motion, the use of pixels in close proximity to the missing macroblocks yields the best results. Also, it is clear that increasing pixel depth is not always an improvement; computational complexity can be decreased, while maintaining a good measure of quality as defined by PSNR. Lastly, the use of the motion vectors generated by the encoder even without accompanying DCT error information can greatly improve the performance. Temporal concealment can be an effective tool in error concealment; however, in sequences with complex motion, during scene changes and when concealing previously occluded areas, spatial concealment can often provide better results. The next chapter presents an improved spatial error concealment algorithm using Hough transforms.

CHAPTER IV

USE OF THE HOUGH TRANSFORM IN SPATIAL ERROR CONCEALMENT

4.1 Introduction

During the transmission of video, errors can cause the loss of single or multiple macroblocks in block-based, motion-compensated temporally predicted, transform-coded video. Although there are many ways to correct these losses, the use of spatial concealment, which relies on data from surrounding macroblocks in the same frame, is always a valuable scheme. Since most spatial concealment schemes consist of interpolating the data from the adjacent blocks [2], it typically works well in areas of little texture with no significant edges, and conversely very poorly in areas containing edges or significant texture. This chapter describes a spatial concealment scheme based on image segmentation followed by directional filtering or interpolation. The best direction for filtering is based on the Hough transform. The results show a significant improvement over the use of bidirectional interpolation especially in the area of significant edges and texture.

In the next section we will review spatial error concealment followed by a review of the Hough transform in Section 4.3. Section 4.4 will present a description of our error concealment algorithm using the Hough transform and Section 4.5 will discuss the results, followed by a brief conclusion.

4.2 Spatial Error Concealment

In any error concealment algorithm, spatial error concealment must be implemented. Data lost in discarded packets or before resynchronization must be replaced and temporal concealment is not always available because of occlusion or scene changes. Others have used a spatial concealment scheme based on bidirectional interpolation of the boundary pixels of the surrounding blocks. The missing pixels are estimated as weighted averages of four

neighbors (Figure 15 (a), Equation 4) where N is the width of the block, λ is $\frac{1}{2}$, and μ_n is the distance operator defined in Equation 5.

$$x_{i,j} = \lambda(\mu_i x_{i,-1} + (1-\mu_i)x_{i,N}) + \\ + (1-\lambda)(\mu_j x_{-1,j} + (1-\mu_j)x_{N,j}) \tag{4}$$

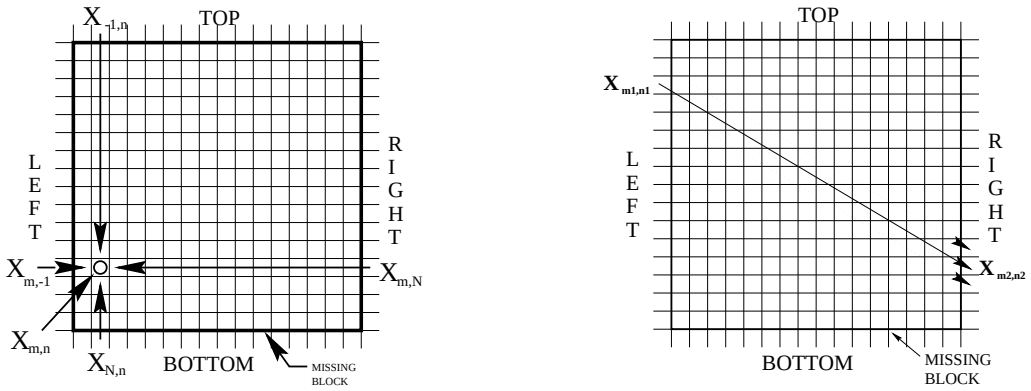
$$\mu_n = 1 - (\frac{1}{2N} + \frac{n}{N}) \quad \text{for } 0 \leq n \leq N-1 \tag{5}$$

However, because of the low pass nature of interpolation, this leads to a blurred appearance and a smearing of edge features [2] [16]. Furthermore, simple interpolation does not exploit any correlation this block may have with its neighbors.

An alternative to bi-directional interpolation is directional interpolation which is illustrated in Figure 15(b). The values along the direction of interpolation are distance weighted as above, but, only in one direction. The value given to pixels is shown in Equation 6 where N is the length of the line between edge pixels x_{m_1,n_1}, x_{m_2,n_2} and μ_n is a linear distance measure similar to the one defined in Equation 2.

$$x_{i,j} = (\mu_i x_{m_1,n_1} + (1-\mu_i)x_{m_2,n_2}) \quad \forall x_{i,j} \text{ on the line between } x_{m_1,n_1} \text{ and } x_{m_2,n_2} \tag{6}$$

Others have used directional interpolation to improve the edge features but have limited the number of directions. Suh and Ho [51] use the boundary pixels of the missing block to determine in which of four directions to interpolate. Kwok and Sun [28] used a Sobel gradient operator and a voting scheme over the pixels surrounding the macroblock to choose from eight possible directions. Their directional interpolation scheme was similar to the one above with the exception of using more than the first edge value, but with the same type of distance operator. The use of directional interpolation can improve the reconstruction of edge features in missing macroblocks; however, selecting the correct direction to interpolate is critical. In the next section the use of the Hough transform to determine the best direction for directional filtering or interpolation is presented.



(a) Bidirectional Interpolation. (b) Directional interpolation.

Figure 15: (a) Bidirectional Interpolation: Distance weighted interpolation using four nearest neighbors. **(b)** Directional interpolation: Distance weighted interpolation between two edge pixels in specified direction.

4.3 The Hough Transform

Duda and Hart[14] pioneered the use of the Hough transform to determine the location of line segments in an image as an application of synthetic vision. This work uses the Hough transform to gain information about edges and lines contained in binary images and uses this information to determine the best direction for filtering or interpolation. An overview of the Hough transform and its use in line detection will be presented next.

The line $y = mx + b$ can be expressed in polar coordinates as in Equation 7

$$\rho = x\cos(\theta) + y\sin(\theta) \tag{7}$$

where (ρ, θ) define a vector from the origin to the nearest point on the line. If we consider the two-dimensional space defined by ρ and θ, any line in the x, y-plane plots to a point in ρ, θ space. Therefore the Hough transform of a line in x, y space is a point in ρ, θ space.

Now consider a particular point (x_1, y_1) in the x, y-plane. There are an infinite number of lines that pass through this point, and each of these lines plots to a point in ρ, θ space. All of these possible lines must satisfy Equation 7 with (x_1, y_1) as constants. This locus of all such lines in the x, y-plane is a sinusoid in the ρ, θ plane. In a dual sense all points on a line in x, y-plane must map to a single point in the ρ, θ-plane. This is shown in Figure

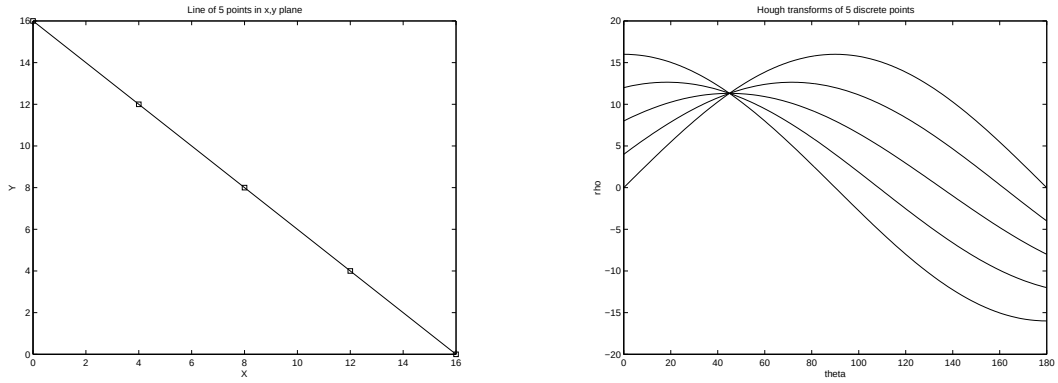


Figure 16: These two figures show how 5 discrete points along a line in x, y space plot to the same point in ρ, θ space.

16 where the left graph shows a line consisting of 5 discrete points, while the right shows the 5 sinusoids corresponding to all possible lines through these 5 points. Note that all the sinusoids cross at the point $(\frac{\pi}{4}, 8\sqrt{2})$ since this defines the line on which all the points lie.

Thus to find the straight line segment upon which a set of points falls, we can set up a histogram in ρ, θ space. For each point in x, y-space we plot ρ for $0 \leq \theta \leq \pi$ as defined in Equation 7 and increment the appropriate bins in the ρ, θ histogram. This corresponds to all possible lines passing through that point. After completing all points on a line, the bin containing (ρ_0, θ_0) will be a local maximum. Thus a search of ρ, θ space for local maxima will yield the parameters for line segments.

This tool has an advantage over other directional determination schemes in that it does not only find the most predominant direction, but is capable of finding all the possible lines located in a given space. The next section will describe the error concealment scheme based on the Hough transform.

4.4 Directional Error Concealment Using Hough Transforms

The previous section illustrated the use of the Hough transform to determine edge and line features in a binary figure. This section presents the algorithm used to perform error concealment based on directional filtering and interpolation. A block diagram is shown in Figure 17.

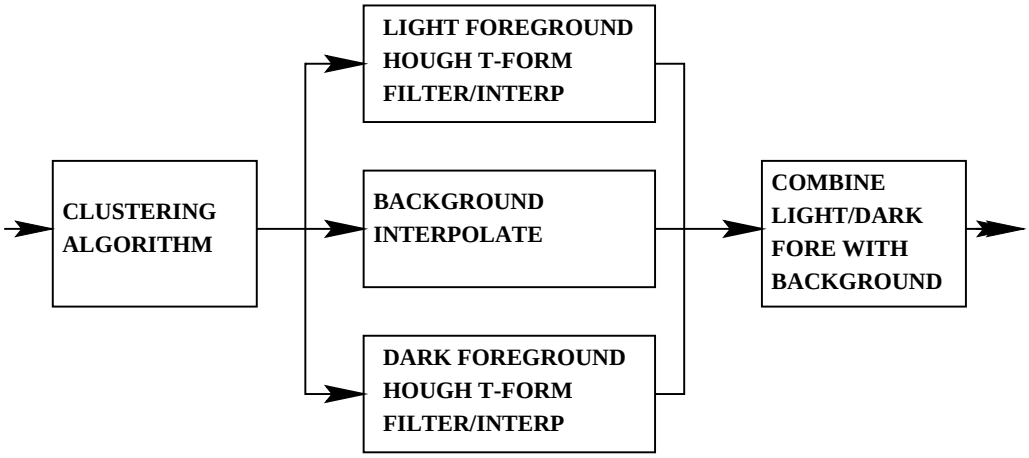
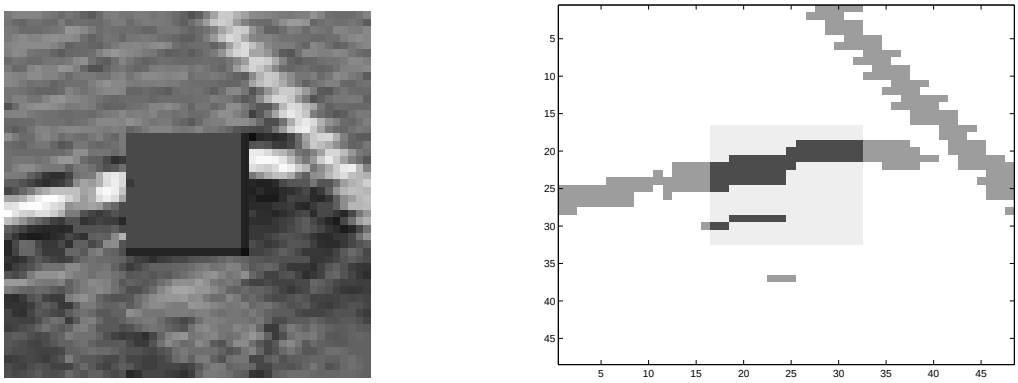


Figure 17: Error-concealment system using the Hough transform and directional filtering/interpolation.

Hough transforms work with binary pixels. Therefore, the image must be segmented first. The algorithm progresses as follows. Begin with a 3×3 macroblock area centered over the missing macroblock. The size is normally 48×48 pixels, but may be $\frac{1}{4}$ that size if dealing with a down-sampled chrominance (e.g. 4:2:0 format) image. Following the work of [28], the image is segmented into three pixel intensities corresponding to bright foreground, dark foreground and background. This is done with a simple clustering algorithm which yields three binary images. Each image will show the location of pixels categorized as bright foreground, dark foreground and background, respectively.

The background is filled in using bidirectional interpolation as described in Section 4.2. Next, the Hough transform algorithm is used to determine possible lines in both the light and dark foreground images independently. Using the Hough transform, the selection of more than one line is possible. In this algorithm, the first possible line (the largest maximum of the ρ, θ space) is selected and the algorithm determines if the line intersects the missing macroblock. If so, this direction is used for filtering or interpolation. If not, the second possible line (the next largest maximum) is evaluated to determine if it is a better choice. At present this algorithm only looks at the two largest maximums, but more decision making is possible with the data from the Hough transform.

At this point, an example will be illustrative. First a 3×3 macroblock area centered over the macroblock to be concealed (48×48 pixels)(Figure 18(a)) is input to the clustering



(a) Original image.

(b) Binary image.

Figure 18: Nine macroblock segment with missing middle block**(a)**. In binary image on right **(b)**, light gray are original pixels and dark gray indicate pixels filled in by directional concealment.

algorithm which will place each pixel in one of three groups corresponding to background, light foreground and dark foreground. After clustering, two binary images are created to denote which of the pixels are in the light foreground and which are in the dark foreground. The image for the light foreground is shown in Figure 18(b). Note this image is gray scaled to highlight different regions, but the image used in the Hough transform algorithm is binary; i.e. a pixel is present or absent. Using each pixel the program determines the possible lines through the pixel and increments the histogram as appropriate. The resulting histogram is shown in Figure 19, which has been smoothed for improved presentation. The two distinct peaks correspond to the two possible lines in the figure. The algorithm looks at the first peak, corresponding to the line in the upper right corner, and determines that it does not intersect the missing macroblock, so it chooses the second line, through the middle, as the best direction for concealing the error. In Figure 18(b), the light pixels in the surrounding macroblocks represent the pixels selected as bright foreground by the clustering algorithm, while the darker pixels in the concealed macroblock represent those created by the directional concealment algorithm.

After the direction is selected there are two possible ways to correlate the data from the surrounding MB's to the missing block. The first is linear interpolation of the edge points and the second is low-pass filtering in both directions. Linear interpolation takes the edge

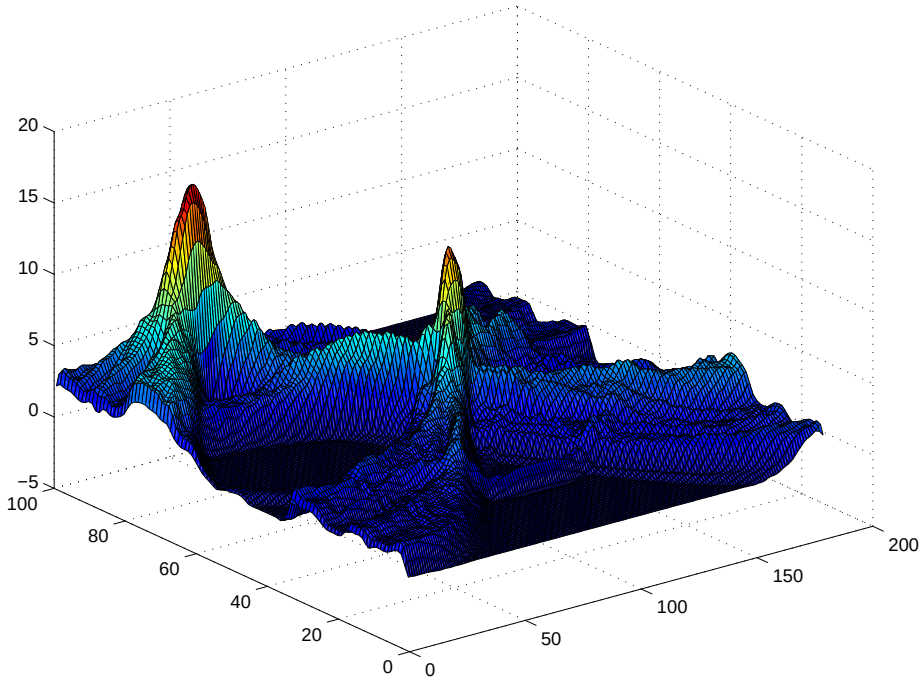


Figure 19: Histogram of Hough transforms of the pixels in Figure 18. Note: histogram has been filtered to improve presentation.

points along a direction and interpolates the missing data. This is shown in Figure 15(b) and Equation 6. In the lowpass filter method, the algorithm selects all the pixels from the 3×3 macroblock in the selected direction (Figure 20) and filters this one-dimensional signal in both directions. The filter is a one-dimensional FIR low pass filter with N = 16. The results are less dependent on the sharpness or cut-off frequency than on the filter length, since the filter delay is used to fill in the missing region.

In the concealment step, the pixels are filled in by the filter/interpolation scheme (Note: not all pixels are filled in) for both the light foreground and the dark foreground. In Figure 18 the pixels filled in by the filtering are black.

After completing the light and dark foreground pixel concealment in separate computations, the three types of pixels, light, dark and background are combined using the following priorities: 1) fill with light foreground if available, 2) fill remaining with dark foreground if available, 3) fill all remaining with background.

Although the above algorithm performs well, in flat areas it is unnecessary because bidirectional interpolation will provide satisfactory results. These regions are recognized by

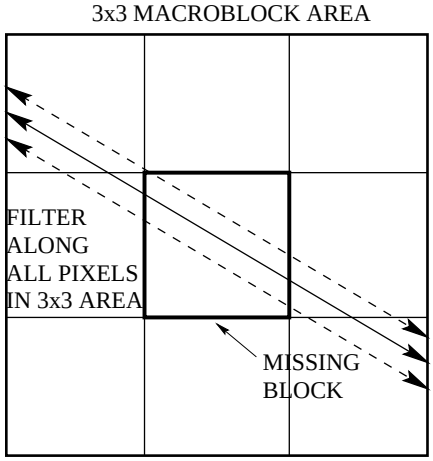


Figure 20: Directional filter is accomplished in both directions along the entire 3×3 macroblock area. Total size is 48×48 pixels.

the algorithm when the clustering algorithm places all the pixels in a single bin. At this point the algorithm exits the directional determination phase and uses bilinear interpolation on all pixels.

In the next section we will discuss the results of the directional concealment algorithm and contrast it to bidirectional interpolation.

4.5 Results

In this section the results of using both directional interpolation and directional filtering will be presented and contrasted with bilinear interpolation. The FLOWER GARDEN sequence was chosen because of the difficult problems it presents. It contains sharp edges and lines, smooth areas of sky, and the highly textured flower garden. Smooth areas are well concealed by any error concealment scheme; however, as will be discussed below, the findings show that techniques presented here help preserve texture and edges features.

The original FLOWER GARDEN frame and the missing macroblocks are shown in Figures 21 and 22. The error concealed figures are shown in Figures 23 thru 25. Table 3 contains the PSNR for the three color components for each figure.

In Figure 23 the neighbor interpolation scheme is shown. Note the inability to maintain edges and the blurred appearance in textured areas. These are the artifacts that this algorithm reduces.

Table 3: PSNR: FLOWER GARDEN for uncorrected, bilinear interpolation, directional interpolation and directional filtering.

	Y	U	V
Error	14.62	16.28	15.82
Bidir. Interp.	24.99	35.82	43.20
Dir. Interp.	26.55	36.30	43.16
Dir. Filter	25.82	36.33	43.20

Figure 21: Original: FLOWER GARDEN.

Figure 22: Error: FLOWER GARDEN with missing blocks.

Figure 23: Bilinear Interpolation: FLOWER GARDEN after error concealment using bilinear interpolation.

In Figure 24 the directional information from the Hough transform is used with a linear interpolation of the edge pixels along the line through the block in the direction selected by the Hough error concealment routine. Note the improved preservation of the edges especially notable around the tree; however, the textured areas have a streaked appearance due to the directional interpolation.

In Figure 25 the same directional information from the Hough transform is used, but in this case the directional line is filtered, in both directions, using long ($N = 16$) FIR low-pass filters. As mentioned above, the cut-off frequency is not a critical parameter, but filter length is, because the time lag in the filter is used to extend the data into the lost macroblock. This technique works well in the textured areas as noted in the actual flower garden portion of the figure. It also preserves the edges very well if the correct direction for interpolation is chosen. Its only drawback is its ability to project undesirable features into the concealed macroblock. For example, note at the long cable on the right side of the picture as it crosses the front right corner of the middle house. The algorithm is able to correctly choose the direction of interpolation to rebuild the lost cable; however the white trim of the house is reflected inside the block inappropriately. In general the use of directional filtering

Figure 24: Directional Interpolation: FLOWER GARDEN after error concealment using directional interpolation.

with the Hough transform produces the best results, and improvements in texture and edge preservation are beneficial in contrast to the occasional artifact it produces.

With respect to the computational complexity, the system was implemented in Matlab and the complexity of the Hough transform is extremely high, but there are many opportunities for reducing this load. First, the transform of the pixels was done over one degree intervals; however when dealing with a 16×16 block, the number of discrete angles is limited which offers a significant opportunity for reductions. Similarly, searching the two dimensional histogram for maxima is computationally intense; however, only direction for filtering is of interest. Therefore, projecting the ρ values onto the θ axis and finding the maxima based on the angles (θ) alone is another reduction in computational complexity. After the angle is selected, the ρ corresponding to a given maximum can be checked to ensure that the line selected is inside the missing macroblock. One final note concerning the Hough transform and computational complexity is that this algorithm is well suited for high speed processing because the ρ's and θ's are computed for each appropriate pixel and there are no data dependencies; therefore the computations can be done in parallel or can be vectorized for pipelining. The computational intensity of the algorithm as implemented is

Figure 25: Directional Filtering: FLOWER GARDEN after error concealment using directional filtering.

considerable, but the reductions discussed here make the algorithm realizable in a real-time environment.

One of the most important findings in this work is the ability of the Hough transform to delineate multiple edges and lines. Although, this algorithm only recognized the first or second line as the best choice, there is more information available. Future work in this area will include the possibility of filtering in more than one direction, which could allow the intersection of two strong lines to be correctly reconstructed.

A second area that offers an opportunity for further investigation is the use of the directional information. This work looked at filtering and interpolation, but other possibilities exist. As mentioned above the filtering provides better results in textured areas while interpolation creates fewer edge artifacts. A system that adapts to smooth or textured surfaces may be able to use the best property of each technique.

4.6 Summary

In this chapter the use of the Hough transform for error concealment in video processing is presented. The method used to determine the best orientation for directional filtering or

interpolation provided improved results in all structured areas. In addition, the use of directional filtering provided the improved results in textured areas, but could lead to artifacts along edges. Interpolation did not produce the artifacts but the results in textured areas was not as impressive. Overall, the results in either case were a significant improvement over simple interpolation both in subjective viewing and in PSNR improvements.

CHAPTER V

USE OF STEGANOGRAPHY IN ERROR CORRECTION AND CONCEALMENT

5.1 Introduction

The MPEG-2 compliant codec presented in this chapter uses data hiding to transmit error correction information and several error concealment techniques in the decoder. The decoder resynchronizes more quickly with fewer errors than traditional resynchronization techniques. It also allows for perfect recovery of differentially encoded DCT-DC components and motion vectors. This provides for a much higher quality picture in an error-prone environment while creating an almost imperceptible degradation of the picture in an error-free environment. In the next section a background of steganography is presented, its application to the codec is discussed in Section 5.3, the results are presented in Section 5.4, and Section 5.5 contains conclusions and areas of further research.

5.2 Steganography - Data Hiding

Steganography is the art and science of data hiding. It can take many forms, has many applications and has a rich and interesting history [43] [3]. Data hiding can be used for clandestine transmissions, closed captioning, indexing or watermarking. This section provides an overview and interested readers are referred to [43][25][29].

Steganography can be viewed as a three-tuple of compromise consisting of detectability, robustness and bit rate (see Figure 26 and Table 4) [49]. Detectability is the main concern of clandestine transmission and is often used in conjunction with encryption. Robustness to all types of processing such as transformations, filtering, truncation, and scaling is the primary concern of the watermarking community. Finally, bit rate, or maximizing the amount of data that can be transmitted without serious degradation of the signal (a form of detection) is a concern for those interested in data tagging, indexing and closed captions. Data hiding

Table 4: Trade-offs in the data hiding world.

PERCEPTIBILITY	Detection and Exploitation Signal Degradation
ROBUSTNESS	Unintentional Removal (Compression/Cropping) Filtering and Noise Intentional Removal (Attack) Multiple Watermarks
BIT RATE	Closed Captioning Indexing Clandestine Transmissions

has been accomplished using musical scores, invisible inks, word spacing and many other ingenious methods and its principles apply to data transmission of all types; however, this discussion will be restricted to data hiding in digital signals and more specifically, images and video.

A typical data hiding process (see Figure 27) begins with a signal, X, and a message, M with an option of using a key, K for encryption of the message. After inserting M into X, the resultant signal X_m is then transmitted. During this transmission it may be subject to different attacks ranging from a noisy channel to intentional attempts to remove the message. Using the received signal, $\widehat{X}_m$, the receiver attempts to recover the original message, M, or at least detect its presence. To extract the message, the original signal, X, and a key, K, may or may not be required. First, the requirement to use the original signal to recover or detect the message is limited to the watermarking or clandestine transmission communities where authenticity or proof of ownership is required; however, a key is used in almost all applications. The keys may be public or private and may be as simple as

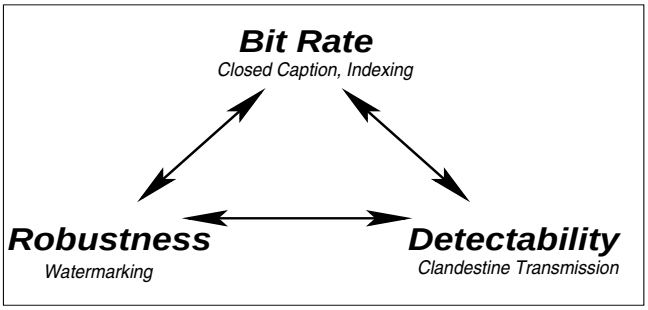


Figure 26: Steganography can be viewed as a triangle of trade offs with considerations for detectability, robustness and bit rate.

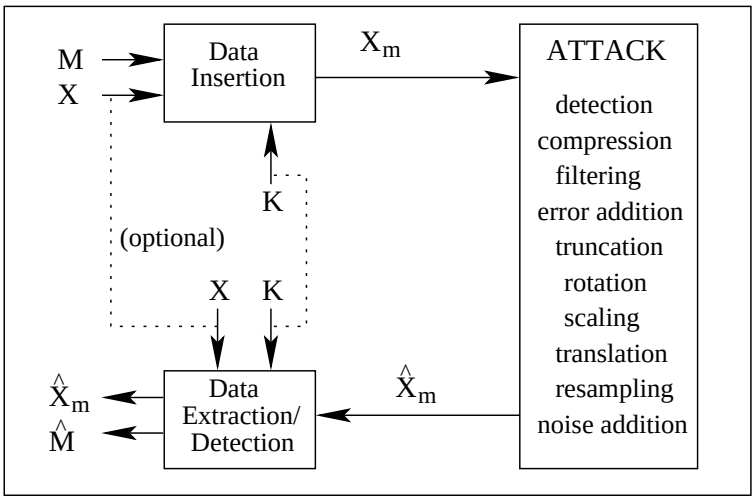


Figure 27: A data hiding system will consist of a host signal X and a message M that are subject to a possible attack. The receiver must be able to decode or detect the hidden message $\widehat{M}$.

defining the location of the hidden data or it may be a cryptographic key to dissuade unauthorized access. Another common example of a key is the spreading sequence used in a spread spectrum data hiding model. This general model covers most modern day data hiding schemes. Next, several examples of data hiding in images and video will be presented. Because of its commercial interest, most data hiding publications are concerned with watermarking; however, many of the concepts remain the same in all applications of this technology.

Data hiding in images and video is usually accomplished with imperceptible modifications to the digital data. In a general sense, data hiding can be segmented into two major divisions, those in the spatial domain and those in the frequency domain. In the spatial domain, the variation of a few pixels whose location is only known by the sender and intended recipient is one technique. An example of this is Patchwork [8] where the key is the location of the altered pixels. This system is robust, but the bit rate is extremely low since only one bit is transmitted per image (the image can be segmented to increase the bit rate). Another is the minor modification of some or all pixels by imperceptible values. This minor modification in conjunction with spread spectrum techniques is a popular method of watermarking [20]. Again, spread spectrum techniques are robust but large chip rates can

lead to very low bit rates. In the spatial domain, imperceptibility can be difficult to attain; therefore many researchers have attempted to use the frequency domain for data hiding.

In the frequency domain, the transform (e.g. FFT, DCT) of the image is taken, and again some or all of the coefficients are altered. Working in the frequency domain has several advantages. First, perceptual models can be used to increase the imperceptibilty of the hidden data. Also the energy spreading of the transform allows the data to be hidden across the entire image, and finally, when working with compressed images or video, variations in the frequency domain can be more easily embedded. The masking characteristics of the human visual system are exploited by Barni et al. [6] in their DCT-domain system for watermarking. The ability to place a spatial watermark in a compressed image is demonstrated in [20] where the authors add a DCT version of their spatial watermark to JPEG compressed images. In this system, watermark detection can be accomplished in either the frequency or spatial domain. Frequency domain techniques are extremely powerful tools in steganography.

Bartolini et al. [7] used steganography to hide error detection information in the DCT coefficients of a low bit rate, block based codec. Although, their error detection scheme degraded MPEG-2 video significantly, the embedding of error detection information was the idea that motivated this approach.

In the codec presented here, frequency domain data hiding is used by modifying the AC coefficients of the DCT to transport error-correction information. Using a technique that toggles the least significant bit (LSB), the bitstream carries error-correction data, while introducing an almost imperceptible degradation of the video quality. The addition of error correction data in this method is preferred over User-Defined data as allowed by the MPEG2 standard, because it distributes the recovery data over the entire file, vice placing it all in, for example the sequence header. This distribution significantly decreases the possibility of losing both a slice and the error correction data. In the next section the data hiding process of the Steganocodec will be described.

5.3 Steganocodec

In previous research in this area, the author determined that the significant source of error during decoding is the loss of synchronization. If the decoder can be rapidly resynchronized, the spatial propagation of errors is significantly limited. In the same research, the author found that after resynchronizing, the loss of differentially encoded DCT-DC values in I frames created very noticeable artifacts. Exact recovery of the DC values is not possible and estimating them requires computationally intensive methods. Similarly, in P and B frames, the loss of the differentially encoded motion vectors also caused significant errors. Given these considerations, the rapid resyncronization of the decoder, the location of the next macroblock (MB), the final DCT-DC coefficients and the final motion vectors are important parameters for limiting error propagation. The ability and method of the Steganocodec to transmit and use these elements will be described in the remainder of this section.

5.3.1 Encoder

Video frames are input to the encoder (Figures 28) and in compliance with MPEG-2 standards, the frames are denoted as I,P or B frames based on the values of N (number of frames in GOP) and M (I/P frame distance). For each frame a slice n, is encoded and the following statistics are collected: the number of bits for each macroblock, the final DC coefficients for I frames, final motion vectors for B/P frames and the number of byte alignment bits at the conclusion of the slice. In the next slice, $n+1$, the data from the previous slice, I_s, is hidden in the DCT coefficients as described next.

The data hiding scheme used by the steganocodec is shown in Figure 29. Initially, the DCT of 8×8 blocks (image information in I frames and error information in P/B frames)

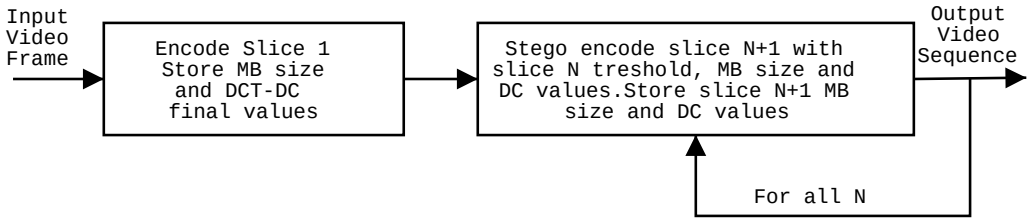


Figure 28: The Steganocodec stores error correction data in the following slice.

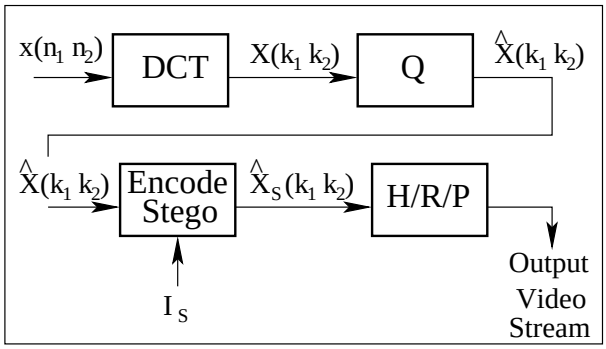


Figure 29: The Steganocodec stores error correction data in the AC-DCT coefficients.

is taken as shown in Equation 8.

$$X(k_1,k_2) = C_{k_1}C_{k_2}\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1} x(n_1,n_2)\cdots$$
$$\cos\left(\frac{(2n_1+1)k_1\pi}{2N}\right)\cos\left(\frac{(2n_2+1)k_2\pi}{2N}\right)$$
$$\text{with}\quad C_{k_1},C_{k_2} = \begin{cases} \frac{1}{\sqrt{N}} & \text{if } k_1=0, k_2=0 \\ \sqrt{\frac{2}{N}} & \text{otherwise.} \end{cases} \tag{8}$$

Next the coefficients are quantized using the quantization matrix $Q(k_1,k_2)$ in Equation 10 (Note: in I frames, the DC component of the DCT is removed and encoded separately).

$$\widehat{X}(k_1,k_2) = \frac{X(k_1,k_2)}{Q(k_1,k_2)} \quad \text{with } k_1,k_2 = 0\ldots 7 \tag{9}$$

with

$$Q = \begin{pmatrix} 8 & 16 & 19 & 22 & 26 & 27 & 29 & 34 \\ 16 & 16 & 22 & 24 & 27 & 29 & 34 & 37 \\ 19 & 22 & 26 & 27 & 29 & 34 & 34 & 38 \\ 22 & 22 & 26 & 27 & 29 & 34 & 37 & 40 \\ 22 & 26 & 27 & 29 & 32 & 35 & 40 & 48 \\ 26 & 27 & 29 & 32 & 35 & 40 & 48 & 58 \\ 26 & 27 & 29 & 34 & 38 & 46 & 56 & 69 \\ 27 & 29 & 35 & 38 & 46 & 56 & 69 & 83 \end{pmatrix} \tag{10}$$

Next, the hidden data I_s, is inserted in the bitstream as shown in Equation 11. Since

I_s is binary (i.e. $I_s = 0, 1$) it acts to toggle the LSB's of the non-zero DCT coefficients, $\widehat{X}(k_1, k_2)$.

$$\widehat{X}_S(k_1, k_2) = \begin{cases} \widehat{X}(k_1, k_2) & \text{if } \widehat{X}(k_1, k_2) <= T \\ \widehat{X}(k_1, k_2) & \text{if LSB}(\widehat{X}(k_1, k_2)) = I_S(k_1, k_2) \\ \widehat{X}(k_1, k_2) + 1 & \text{if LSB}(\widehat{X}(k_1, k_2)) \neq I_S(k_1, k_2) \end{cases} \tag{11}$$

In the encoder, a global thresholding value, T, is used to determine which coefficients are encoded. This gives the encoder the ability to influence the SNR of the bitstream without significantly changing the total bit rate. After some initial testing, we found that if all non-zero coefficients were used, image quality was maintained, with no change in the total bit rate. The final steps in Figure 29 are Huffman and run-length encoding, followed by packetization of the coefficients into a MPEG-2 compliant bitstream.

A few additional notes concerning the encoder are in order. The probability of any MB being corrupted is not uniform. Within a slice, any error causing a loss of synchronization will corrupt that block and all blocks that follow. For example, consider a slice with 45 MB's each containing the same number of bits. If an error occurs randomly in the slice, the probability that the first block is lost is $\frac{1}{45}$ while the probability that the last block will be lost is 1. With this in mind, and the fact that adjacent slices may be lost, one should carefully consider the order in which the data is transmitted. Therefore, the DCT-DC coefficients (I frame) or motion vectors (P/B frame) are transmitted first, followed by the byte alignment offsets and finally the size of the MB's in reverse order (last MB in slice first). This scheme improves the probability that resynchronization data will be available even if adjacent slices are lost.

In this implementation the last slice is unprotected. It is possible to include this data in the first slice of the following frame, but for simplicity of decoding a single frame at a time, it was decided that the last slice would be unprotected. Additionally, errors in this slice are easily concealed because this is rarely a region of interest. Next, the operation of the decoder will be presented.

5.3.2 Decoder

One of the key aspects of the steganography scheme is full compliance with the MPEG-2 standard. In the decoder, all the hidden data may be discarded by a coder incapable of using the data, while its full value is available to a stegano-decoder. The decoder is the inverse of Figure 28(b) which includes retrieval of the hidden data I_s, inverse quantization ($\widehat{X} * Q$), and the inverse DCT to the spatial domain. If an error is recorded in a slice (Figure 30), the data remaining in the slice (up to the next start-of-slice header) is stored until the entire frame is decoded. Next the hidden data is used in an attempt to recover the lost slice. Using the size of the MB's, the decoder can locate the beginning of the next MB to be decoded and the DCT-DC values or motion vectors are computed using the stegano-encoded final value and the differential values. If the data is not available, due to multiple errors in the adjacent slices, then the decoder will attempt to conceal the error using an Early Resynchronization (ER) scheme as described in [30]. Even when ER is used, any available information such as DCT-DC values or motion vectors, is used to enhance the recovery. The data hiding algorithm presented here produced good results, but reduces the PSNR of the received video. The next section will discuss the location and perceptibility of this error.

5.3.3 Error Analysis

An important aspect of any algorithm using steganography is to place the data in the most imperceptible location. The technique used here takes full advantage by using the LSB of

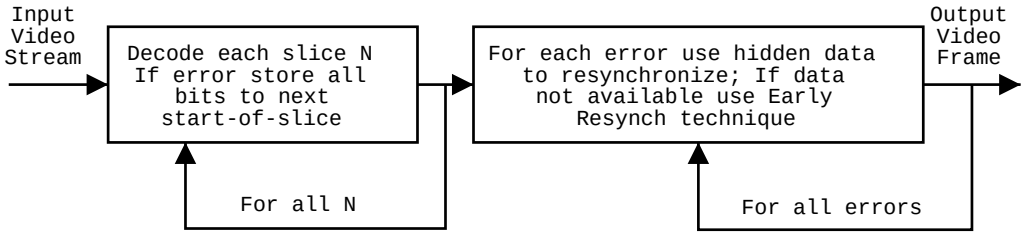


Figure 30: During the decoding process all errors are stored until entire frame is decoded. Then errors are corrected/concealed using the stega data. If stegano data is not available, the Early Resynch technique is used.

the quantized DCT coefficients. In this section a 256×256 version of the CAMERA MAN, Figure 31(a), will be used to illustrate the worst case effects of the data hiding algorithm. In this analysis, the image will be treated as an intra-coded frame by the Steganocodec.

The first error introduced by the by the encoding system in Figure 28 is the quantization error ϵ_Q (Equation 12) which is related to the frequency domain error E_Q by Equation 13. Note that the quantization in Equation 9, multiplication by $\frac{1}{Q}$ in the frequency domain, relates to convolution in the time domain. The values in the quantization matrix, Q (Equation 10), act as a low pass filter as shown in Figure 31(b). Figure 32 is the compressed version of CAMERA MAN using the MPEG2 quantization matrix and the magnitude of the error, $\mid \epsilon_Q \mid$. The error has been scaled for better presentation.

$$\epsilon_Q = x(n_1, n_2) - \widehat{x}(n_1, n_2) \tag{12}$$

$$E_Q = X(k_1, k_2) - \widehat{X}(k_1, k_2) \tag{13}$$

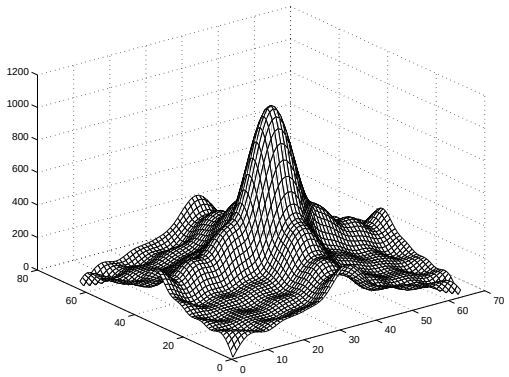
The next error, ϵ_s, is the distortion of the DCT coefficients with the hidden data (Equations 14 & 15). Recall from Equation 11 that I_s has no effect on DCT coefficients below the threshold, T, and will introduce a distortion only if the LSB of the remaining coefficients differs from $I_s(k_1, k_2)$ (approximately 50% of the time). For illustration, the maximum possible error, (i.e. every non-zero coefficient was changed) has been induced in Figure 33 which shows both the image and the magnitude of the error, $\mid \epsilon_s \mid$. Again the error is scaled for viewing. The same scale is used for both Figures 32 and 33 so meaningful comparisons can be made.

$$\epsilon_s = \widehat{x}(n_1, n_2) - \widehat{x_s}(n_1, n_2) \tag{14}$$

$$E_s = \widehat{X_s}(k_1, k_2) - \widehat{X}(k_1, k_2) = \widehat{X}(k_1, k_2) + I_s(k_1, k_2) \tag{15}$$

In the decoder, the inverse quantization, which involves multiplication by Q, will act as a high-pass filter of the signal $\widehat{X_s}$ (Figure 34). High-pass filtering a signal with errors can result in a significant degradation of image quality; however, because of the low-pass

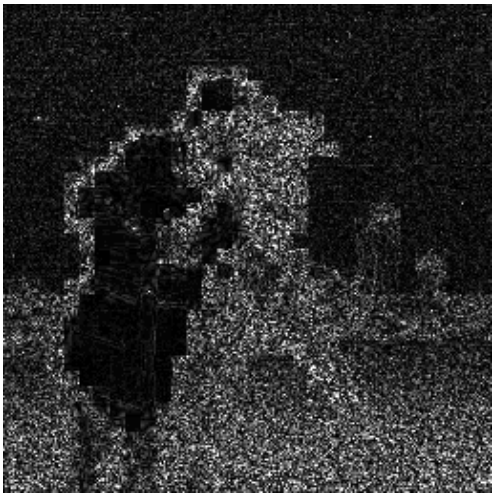
(a) Original.



(b) Frequency Response.

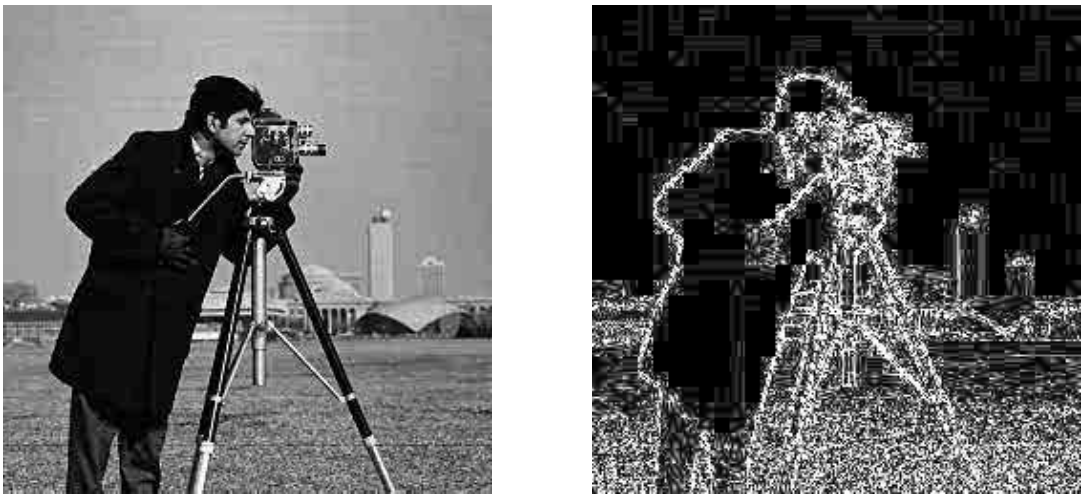
Figure 31: **(a)** Original frame of CAMERA MAN; 256×256. **(b)** This is the frequency response of the quanization matrix, $1/Q$.

(a) Compressed.



(b) Magnitude of Error, $\mid \epsilon_Q \mid$.

Figure 32: **(a)** Compressed version of CAMERA MAN using the MPEG2 quantization matrix. **(b)** Magnitude of Error, $\mid \epsilon_Q \mid$, scaled for viewing.



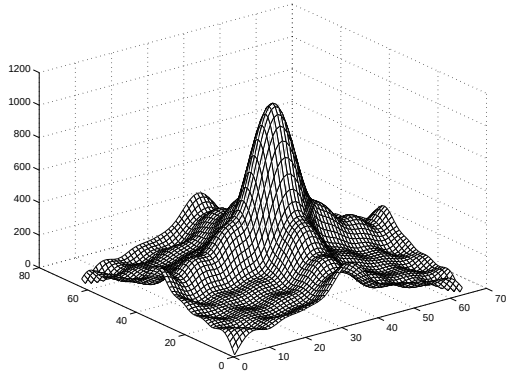
(a) Compressed with hidden data.

(b) Magnitude of Error $| \epsilon_s |$.

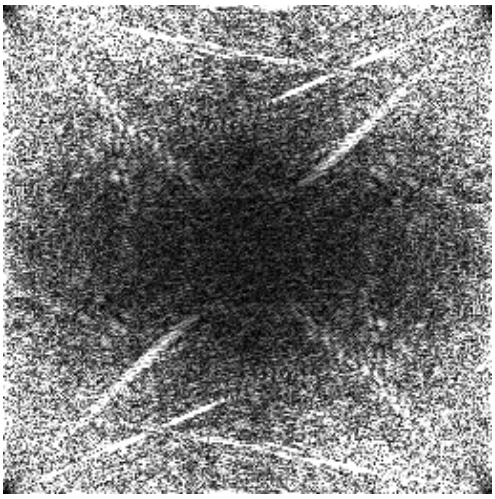
Figure 33: (a) Compressed version of CAMERA MAN with the maximum data hiding error. **(b)** Magnitude of maximum possible error, $| \epsilon_s |$, scaled for viewing.

filtering in the encoder, most of the high frequency components are zero and unaffected. This effect is illustrated in Figures 34.

Figure 33(b) shows the magnitude of the error, $| \epsilon_s |$, introduced by the Steganocodec. Note the error is hidden in the areas with significant texture where it is most difficult to detect visually. Finally, Figure 34(b) shows the magnitude of the 2D-FFT of the error. In this figure, the corners are low frequency with the center showing high frequency in both the horizontal and vertical directions; the figure is symmetrical since all the pixel values are real. The image is gray scale with low values in black. The corners are all black indicating that the DCT-DC coefficients are unaffected. The center is also dark indicating that very little of the error exists in the high frequency portions of the image. The lighter colors show the error is hidden in the middle frequency portion of the image where it is least perceptible. There are two things to notice concerning this example. First, this is the maximum error; i.e. all the non-zero DCT coefficients were changed; and second, this image is only 256×256, much smaller than the typical MPEG-2 video format, which makes the errors appear to be more significant than it was in the video trials presented here.



(a) Frequency Response.



(b) FFT of Error, $\mid \epsilon_s \mid$.

Figure 34: (a) This is the frequency response of the inverse quanization matrix, Q. **(b)**Two dimensional FFT of the error ϵ_s. Low frequencies are in the corners, high frequencies in the center of the image. Black corresponds to low values; white to high; Note the error is located in the middle frequencies in both horizontal and vertical directions.

5.4 Results

In this section, the results of the Steganocodec are presented in the following form. First the impact of the Steganocodec on the image quality in an error-free environment will be shown. Next, with errors introduced into the bitstream, the results of different error concealment schemes will be contrasted with the Steganocodec. Finally, this section will conclude with observations and comments about the data.

As discussed in the previous section, the Steganocodec's ability to carry the additional information results in a degradation of picture quality; however, in an error-prone environment, the improved performance in error correction and concealment compensates for the minor decrease in quality of all frames. Presented below are results for 13 frames of three sequences: FLOWER GARDEN, CHEER, and TENNIS. $(N = 6, M = 3)$ These sequences of 13 frames allow the algorithm to work with a Group of Pictures (GOP) following a scene change (Frame 0) and another without the scene change.

Table 5 shows the capacity of the stegano channel and the loss of image quality with

Table 5: Stegano characteristics for 13 frames of FLOWER GARDEN, CHEER and table TENNIS.

	No Stego (Bytes)	W/ Stego (Bytes)	Data Xfer (Bits)	Y PSNR (dB)
FLOWER	545585	524484	526994	28.74
CHEER	546728	546829	525118	32.94
TENNIS	542158	544453	515725	36.74

the inclusion of the stegano information. The first two columns are the number of bytes to encode the 13 frames without and with the inclusion of stegano information. Note, there is no significant change of the bit rate with the inclusion of the stegano information. Next, the total number of bits available for transmission of stegano data shows that the channel capacity is about 12% of the total channel. This number varies based on the threshold used by the Steganocodec and the bit rate of the encoded stream. Although not shown, the entire capacity of the channel was not used. The number of bits actually used to transmit all required stegano information ranged from approximately 76% to 80%. The last column represents the PSNR of the luminance signal as compared to the same sequence coded without any stegano data. Subjectively, using a high quality 21" TV monitor, we found that a PSNR of 15 presented a snowy picture; when the PSNR was 25, the error was barely noticeable, and a PSNR of about 35 made the errors undetectable. In conclusion, the loss of quality for the three sequences was minimal.

Table 6 shows the PSNR for 13 frames of the FLOWER GARDEN sequence using several different error correction schemes. The bitstream was encapsulated in an AAL5 type packet and packets were lost with uniform probability with a loss rate of 10^{-4}. The first column is simple temporal error concealment, using the information from the previous frame with no correction for motion. The second column uses an ER scheme similar to [30] and last column is error correction and concealment using the Steganocodec. Examples of the ER and Steganocodec sequence are shown in Figures 37 and 38. The improvements of the Steganocodec over the other schemes is the result of two significant factors. First is the importance of the DCT-DC components. In the ER scheme these are estimated from the surrounding macroblocks while in the Steganocodec, they are computed from the final values

Table 6: PSNR for 13 frames of FLOWER GARDEN using Temporal error concealment, ER, and the Steganocodec.

Frame	Temporal	ER	Stego
0	10.33	18.92	22.07
1	10.93	14.85	22.20
2	10.05	19.54	22.10
3	10.09	19.90	23.19
4	10.47	21.12	23.42
5	10.25	22.42	24.88
6	10.25	25.11	28.29
7	10.41	22.38	24.91
8	10.19	24.30	26.42
9	9.87	23.13	26.66
10	10.19	24.67	26.19
11	10.24	25.58	26.58
12	10.32	27.60	27.95

as described in Section 5.3. Second, in early resynchronization, the maximum number of macroblocks is not always decoded correctly; the first macroblocks frequently contain errors that are not detected by the decoder. Both of these issues, although evident in PSNR are much more an issue in subjective viewing since both cause very noticeable artifacts.

A few additional comments are in order. The improvement in the PSNR as the sequence progresses is the direct result of the improvement of the temporal error concealment of lost macroblocks as the scene progresses. Also, the improvement is partly due to the error sequence itself. The first I frame contained 15 errors with a loss of 296 MB's while the last I frame contained only 7 errors and lost 172 MB's.

Again, using data from the FLOWER GARDEN sequence, Table 7 illustrates the performance of this codec. Note the rows define the type of frame, I, P, or B, with the first column indicating the number of each frame type and the total and average number of errors for each. The next column indicates how often the stegano data was recoved and used. The last two columns pertain to the loss of adjacent slices. The first indicates the number of adjacent slices lost in each frame type and the last is the usability of the stegano-data after the loss of the following row. First, note the number of errors per frame decreases with the size of the frame with I being the largest and B being the smallest. This is consistent with the uniformly distributed error model used. Second, the percentage of times stegano-data

Table 7: Error recovery statistic for the FLOWER GARDEN sequence.

Frame Type	Number Frames	Total Errors	Average Errors	Stego Data	Adjacent Slices	Stego Data
I	3	30	10.0	97%	8	100%
P	2	15	7.5	86%	4	50%
B	8	39	4.9	79%	5	40%
Total	13	84	NA	87%	71%	

was used to recover the data is shown. As detailed in Section 5.3.1 the stegano data is used if available, and early resynch is used if not. In I frames the stegano data was available every time with one exception being an error in the last slice (which is not protected). In locations where adjacent rows were lost, the stegano-data was still available. This can be attributed to encoding the data from the beginning of the slice, decreasing the possibility of loss, the reverse encoding of the MBA's and the high bit rate of I frames. This high bit rate allows the stegano-data to be encoded in the first portion of the slice, again decreasing the probability of loss. In P frames, the performance of the stegano-data is excellent; in the two cases where stegano data failed, it was the result of adjacent slices being lost. In B frames, the performance falls because of the limited bandwidth for the transmission of stegano-data. In this simple panning sequence, the P frames are encoded very efficiently with numerous skipped MB's. This decrease in the bandwith makes the recovery of stegano data with adjacent slice losses virtually impossible. In fact, on a small percentage of the slices, there was insufficient bandwidth to send all the stegano data, which made the recovery of long slice errors not possible. As a closing note, it is most important to protect the I frames and P frames since they impact more than a single frame, and the B frame can be slightly neglected since its duration is short lived. Also, in adjacent slice errors, although MB data was not available, the differentially encoded MV's and DCT-DC coefficients were used to improve the performance.

Figures 35 through 38 are selected results of the ER and Steganocodec algorithms. Figure 35 is the original frame and Figure 36 illustrates the lost slices with no error correction. Figure 37 contains the state of the art resynchronization algorithm and 38 demonstrates resynchornization using the Steganocodec. In contrasting the two results, the importance

Table 8: PSNR for 13 frames of FLOWER GARDEN, CHEER and TENNIS.

	Temporal	Resynch	Stego
FLOWER	10.28	22.27	24.99
CHEER	13.34	18.82	18.98
TENNIS	15.19	24.13	28.07

of accurately recovering the DCT-DC components is clearly evident. In addition, the effects of poor resynchronization is evident in the top frame of Figure 37. The third error (the seventh slice) is an example of the ER routine decoding MBs incorrectly, but without an error being detected by the decoder. Overall, the improvements in both PSNR and subjective viewing are easy to recognize. Although space does not allow, all the sequences in Table 8, show similar improvements in PSNR for the BICYCLE and TABLE TENNIS sequences.

5.5 Conclusions

The Steganocodec uses steganography to transfer resynchronization data from the encoder to the decoder. This hidden information results in an imperceptible decrease in picture quality but does not influence the total bit rate. In all cases, the decoder performed better than early resynchronization.

Topics for further research in this area include the possibility of using a variable rate for the stegano encoder. This may decrease the influence the decoder has on the frames with no errors. Also, adapting the quantity of error correction information, based on the quality of service would allow the best possible picture for a given channel.

In closing, the Steganocodec provides a marked improvement over the existing methods of error correction and concealment while remaining compatible with the MPEG-2 standard.

Figure 35: Original I frame from FLOWERS sequence.

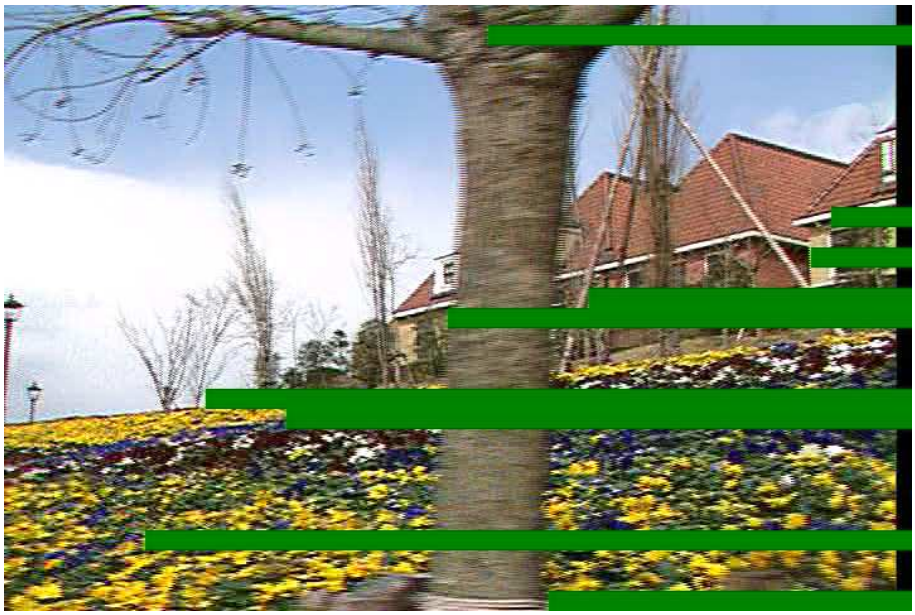


Figure 36: I frame from FLOWERS sequence with no error correction.

Figure 37: I frame from FLOWERS sequence using Early Resynchronization.

Figure 38: I frame from FLOWERS sequence using Steganocodec.

CHAPTER VI

USE OF PARITY INFORMATION TO ENHANCE ERROR DETECTION AND CONCEALMENT

6.1 Introduction

The previous chapter presented the Steganocodec that resynchronizes using the length of each macroblock (MB) and recovers differential DCT-DC components and motion vectors using the final value. The block size and the final values are transmitted using data hiding techniques within the DCT coefficients. Although this system performs well, the Parity-codec extends this work by providing a bitstream with improved error-free performance and with similar performance in an error-prone environment.

As noted in the discussion of Early Resynchronization (Chapter 2), as the error string is decoded, the probability of decoding an error-free macroblock increases as more bits are decoded because end of block indicators in the DCT's and bit stuffing at the end of each macroblock assist the decoder in regaining synchronization. Conversely, the first several macroblocks have the highest probability of being valid decodes that contain errors. Detecting these errors improves the resynchronization process; hence, the codec described in this chapter embeds parity information in the DCT coefficients. This information reduces the probability of a valid but erroneous decoding and enables the decoder to recognize an invalid string more quickly.

Also discussed in Chapter 2, the incorrect recovery of differentially encoded values such as the DCT-DC values in I frames and motion vectors in predicted frames can create noticeable artifacts. Exact recovery of these values is not always possible and estimating them requires computationally intensive methods that are not always effective. For these reasons, this encoder includes the final DCT-DC coefficients for I frames and the final motion vectors for P frames and B frames as user-defined data. This allows perfect reconstruction

of these values. The next section will discuss the encoder followed by a presentation of the decoder in Section 6.3. Section 6.4 will present the results and a summary is contained in Section 6.5.

6.2 Encoder

The Paritycodec uses an information hiding scheme that is compliant with MPEG-2 standards. Similar to the Steganocodec (Chapter 5), raw video is input to the encoder and the frames are denoted as I, P or B frames based on the values of N (number of frames in GOP) and M (I/P frame distance). Each frame is divided into slices, then 16×16 macroblocks, and finally into 8×8 DCT blocks. The DCT of these blocks (image information in I frames and error information in P/B frames) is computed and the coefficients of the DCT are quantized. This step introduces quantization error in the frequency domain error, E_Q, which is related to the spatial domain error, ϵ_Q, by the DCT.

After quantization, the DCT coefficients are transformed from a matrix to a vector using the zigzag scan as shown in Figure 39(a) and Equation 16. Similarly, the quantization error is mapped as shown in Equation 17.

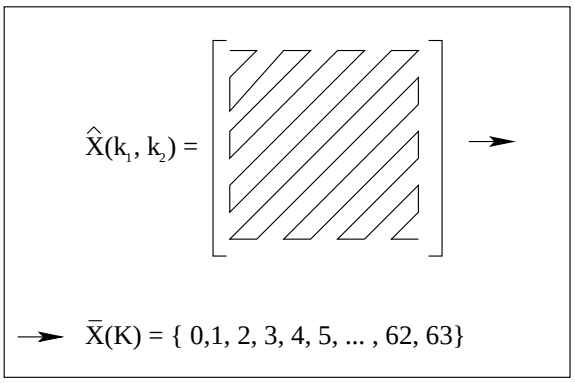
$$\widehat{X}(k_1, k_2) \mapsto \overrightarrow{X}(K) \text{ for } \{K : 0 - 63\} \tag{16}$$

$$E_Q(k_1, k_2) \mapsto \overrightarrow{E_Q}(K) \text{ for } \{K : 0 - 63\} \tag{17}$$

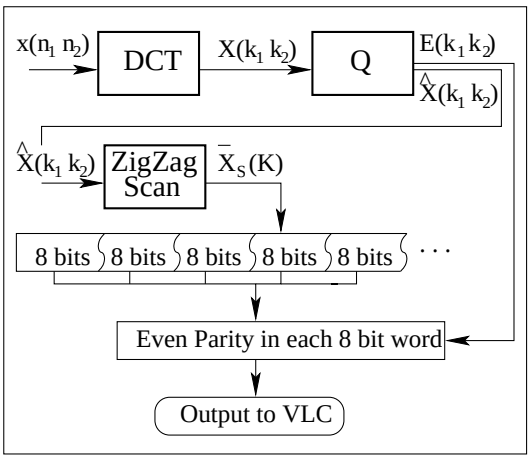
This vector is partitioned into bytes and is forced to have even parity, thus encoding parity information into the bitstream at no additional cost (Equation 18 and Figure 39(b)).

$$\overrightarrow{X}_S(K) = \begin{cases} \overrightarrow{X}(K) & \text{if } \sum_{K=0}^{7} \overrightarrow{X}(K)\%2 = 0 \text{ for K} = 0 \ldots 7 \\ \text{otherwise} & \\ \overrightarrow{X}(K) + 1 & \text{if } \overrightarrow{E}(K) \geq 0 \text{ for } K = \min\ \overrightarrow{E}(K) \\ \overrightarrow{X}(K) - 1 & \text{if } \overrightarrow{E}(K) < 0 \text{ for } K = \min\ \overrightarrow{E}(K) \\ \overrightarrow{X}(K) & \text{for all other } K \end{cases} \tag{18}$$

To elaborate, if a given byte has even parity no change is required; however, if the byte has odd parity, a single bit is changed to force even parity; the bit selected minimizes the



(a) Zigzag.



(b) Data Hiding Process.

Figure 39: **(a)** The 2-D DCT matrix is transformed to a 1-D vector using the zigzag scan. **(b)** Parity information is encoded to minimize the quantization error, $E_Q(k_1, k_2)$.

quantization error, $\overrightarrow{E_Q}(K)$.

After each macroblock in a slice is encoded (Figure 40), the final DCT-DC coefficients for I frames and the final motion vectors for B frames and P frames are stored and transmitted as user-defined data at the beginning of the following slice. The final slice in the image does not receive this protection. This simplification removes the requirement for a two-pass encoder, or error correction across page boundaries. The decoder's use of this parity and user-defined data is presented next.

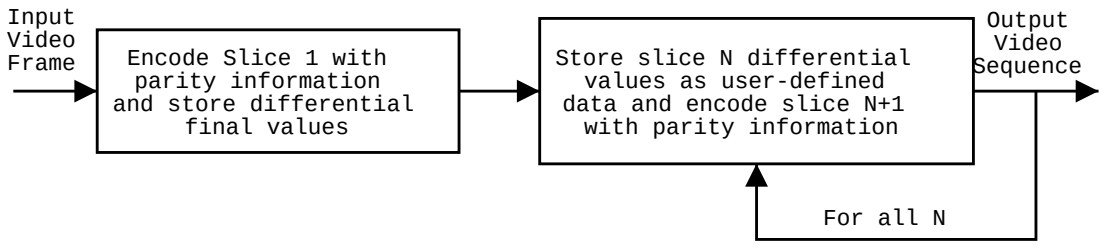


Figure 40: Differential values are transmitted as user defined data, while parity information is encoded in the DCT-AC coefficients.

6.3 Decoder

A key aspect of this steganography scheme is the full compliance of the decoder with the MPEG-2 standard. If the decoder is incapable of using the data, it is ignored; however, the data is available to any capable decoder. The decoder (Figure 41) processes the input data stream until an error is detected. The error may be indicated by a decoder fault, an error start code inserted by a lower level in the transmission process, or as detected by the DCT parity checking. After an error is detected, all bytes up to the next start code are saved in a buffer for later processing. Decoding continues after this start code which is normally the beginning of the next slice. This process continues until the end of the frame, where each error is addressed by the error recovery and concealment scheme which is presented next.

After the entire frame is decoded, each slice containing an error is passed to the error concealment algorithm shown in Figure 42. First Early Resynchronization (ER)(Figure 7) is attempted; however, in this decoder, the ER is enhanced by the parity information in the DCT's. After the Early Resynchronization algorithm is complete, the differentially encoded DCT-DC values (I frame) or motion vectors (P frame, B frame) are recovered after which the bitstream is decoded with the correct initial values, and written to the current frame buffer. The final step recovers any lost macroblocks. Depending on the type of error and the amount of data lost, the algorithm will conceal lost MB's using motion-compensated temporal error concealment. This procedure is accomplished for all slices containing errors.

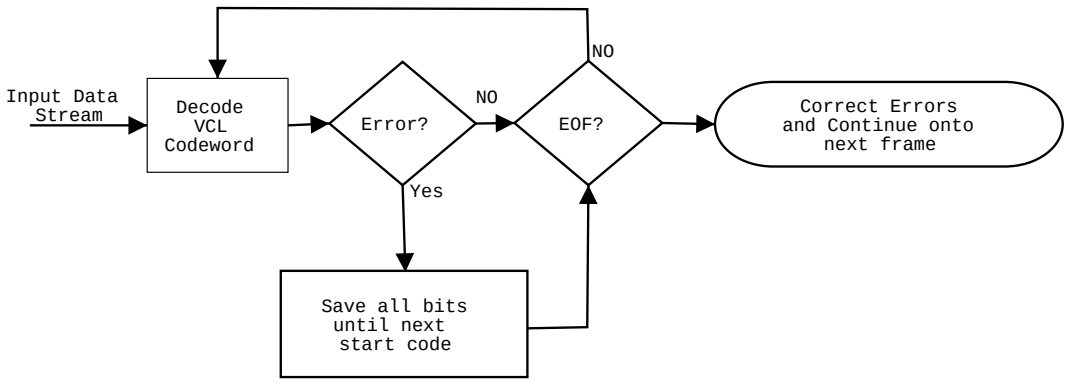


Figure 41: During the decoding process all errors are stored until the entire frame is decoded. Errors are then corrected/concealed using the resynchronization algorithm.

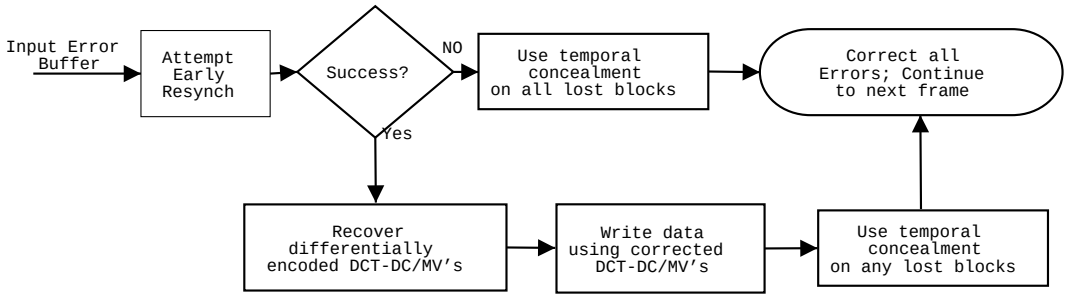


Figure 42: After each frame is decoded, errors are concealed using a combination of Early Resynchronization and temporal error concealment.

The process described here resulted in improved resynchronization of the bitstream following errors as presented in the next section.

6.4 Results

The Paritycodec's ability to resynchronize the bitstream with fewer errors is demonstrated in this section as follows: first, the results with no errors are presented; this is followed by the performance of the decoder in an error-prone environment for I frames, P frames and B frames; and finally, the results of variable length parity words is discussed. Results are presented for three, commonly used sequences: FLOWER GARDEN, CHEER, and BICYCLE.

A major design concern of this codec was to minimize the negative impact of data hiding on the picture quality while still improving the performance of the decoder in an error-prone environment. This impact is shown in Table 9 where the first column is the Y PSNR for the three CCIR 601 sequences encoded at 10 Mbits/sec. The second column shows the Y PSNR for the sequence containing the hidden parity data and the final column shows the same sequences, but with all parity data transmitted as side information. The side information, S (Equation 19)

$$S = (N \times CC \times \frac{9}{8}B + V) \times F \tag{19}$$

equals the number of macroblocks per frame, N, times the number of DCT's per macroblock, CC, (this varies from 6 for 4:2:0 chroma format to 12 for 4:4:4 chroma format), times the number of parity bits per macroblock, B, times $\frac{9}{8}$ to account for the side information overhead, plus the number of bits of final value information (DCT-DC and MV's) per frame,

Table 9: No-error characteristics for 13 frames of FLOWER GARDEN, BICYCLE and CHEER.

	No Steg Y PSNR(dB)	W/ Steg Y PSNR(dB)	Side Info Y PSNR(dB)
FLOWER	31.41	30.90	29.70
CHEER	37.06	36.26	35.09
BICYCLE	33.41	32.73	31.44

V. This is multiplied by the frame rate, F. The value of V equals the number of slices times the number of bits for the final values which varies from $12 - 16$ bits for DCT-DC values in I frames up to 24 bits of vector information for B frames. Typical values for the sequences used here (CCIR 601 30 frames/sec) are 2.19 Mbits/sec for the parity information and 21.6 Kbits/sec for the final values. Side information has a deleterious effect of reducing the number of bits available for encoding and is expressed as bits/sec. The results show that with respect to PSNR, the data hiding system outperforms a similar system using side information to transmit the data. Subjective visual inspection of the sequence shows excellent image quality for all three sequences.

Results for the error-free environment are positive, but the enhancement of this system is most realized in an error-prone environment. In an effort to evaluate the performance of the proposed system, errors are inserted in every slice (except the first and last) in each frame type. Although this is different than typical evaluations in which a bit error rate is selected and the errors are introduced based on a channel model, this allows a more thorough comparison between the current state-of-the-art and the proposed system. In essence, the effort is to evaluate the improvement in the Early Resynchronization algorithms of the two recovery systems, not error concealment in general.

In Tables 10 through 12, the results for the recovery of the sequence following an error is shown. The sequences are 13 frames consisting of 2 groups of pictures (GOP) (M=6,N=3) and contains three I frames, two P frames and eight B frames. Errors are introduced in every slice except the first and last (prevents total failure of the decoder) which provides 28 errors per frame. The errors are placed at the midpoint of the slice, based on the total number of bits used, to allow for multiple resynchronization attempts and errors of sufficient length to evaluate the recovery process. The tests were conducted using two

Table 10: Recovery performance for three I frames in a 13 frame sequence of FLOWER GARDEN, BICYCLE and CHEER.

	Number of Error	SOA Err-Free	Parity Err-Free	SOA Err Blks	Parity Err Blks	SOA PSNR	Parity PSNR
FLOWER	84	6	43	1.81	0.76	19.58	27.28
BICYCLE	84	0	36	2.27	1.25	27.12	36.51
CHEER	84	0	19	3.23	1.99	34.62	49.21

different assumptions. The first set of tests simulated a loss of synchronization caused by a bit error, where resynchronization is typically accomplished very quickly, with the loss of no more than one macroblock. The second set of tests simulated the loss of a packet in a packetized streaming system by discarding 250 bits of data. Results from the two scenarios are very similar except the number of irrecoverable blocks increases in the 250 bit discard tests since more data is lost. The results presented here are for the 250 bit discard scenario.

6.4.1 I frames

Table 10 contains the results for the 13 frame sequences. In this section, state of the art (SOA) refers to a decoder with no data hiding capabilities, but with all other resynchronization capabilities discussed in Section 6.3. Parity is the name used to represent the results from the proposed system. Although the use of the DCT-DC values is part of this work, it is included in both systems; there are several DCT-DC recovery schemes that work reasonably well visually, but small errors would skew the PSNR results too heavily in favor of the parity system.

In Table 10, the first three columns show the number of errors, followed by the number of error-free decodes for the two systems. Error-free decodes refer to a resynchronization of the bitstream which recovers all available macroblocks (some may be missing because of discarded bits) with no errors in the reconstructed frame. As discussed earlier, synchronization with the bitstream improves during decoding because of end-of-block indicators in the DCT and bit stuffing at the end of the macroblocks. Resynchronization results are improved by minimizing the number of blocks in error and minimizing the error in these blocks. The next two columns show the average number of blocks in error in each recovery, and the last

two columns showing the average PSNR of the macroblocks in error. These three metrics, error-free decodes, number of blocks in error, and average PSNR of macroblocks in error, indicate the improvements offered by the Parity system.

As shown by the results, the Parity system shows significant improvements in every metric. The system is capable of finding significantly more error-free decodes. This results from the decoder using the hidden parity information to eliminate incorrect decodes and continue searching for the correct decoding sequence. When an error does occur, the number of blocks in error is fewer and the PSNR of those blocks is higher indicating a less perceptible error. The results of the I frames are very promising because the number of non-zero DCT coefficients is significant enough for the parity system to catch most errors. As shown in the next Section, as the number of non-zero DCT coefficients decreases in the P and B frames, the results are still an improvement; however, not as significantly as shown here.

6.4.2 P and B frames

The results for P and B frames are shown in Table 11 and 12. These results are presented using the same metrics as the I frame with the number of perfect decodes followed by the number of blocks in error and the average PSNR of the blocks in error. In the P and B frame results, there is still a significant increase in the number of perfect decodes and the number of blocks in error is also improved; however, the increases in the PSNR are not as significant. This results from slight errors in the recovery of the motion vectors, that while not significant in viewing can have a significant effect on the PSNR. Similar to the recovery of I frame DCT-DC coefficients, the motion vectors are recreated using a final value (transmitted as user defined data) and the differential offsets; and as with the DCT-DC coefficients, the initial few differential motion vectors may be in error. Although this system of motion vector recovery provides better results than other systems (nearest neighbor MV for example) it was used by both the SOA and the Parity system, so this large advantage would not skew the results unfairly in favor of the Parity system. Another reason for the smaller margin of improvement is the decrease in the number of non-zero DCT coefficients. This lower number of coefficients gives little for the parity system to work with.

Table 11: Recovery performance for two P-frames in a 13 frame sequence of FLOWER GARDEN, BICYCLE and CHEER.

	Number of Error	SOA Err-Free	Parity Err-Free	SOA Err Blks	Parity Err Blks	SOA PSNR	Parity PSNR
FLOWER	56	1	12	1.70	1.30	20.61	21.71
BICYCLE	56	5	12	1.92	1.48	23.07	23.89
CHEER	56	1	13	1.84	1.25	23.81	23.28

Table 12: Recovery performance for eight B-frames in a 13 frame sequence of FLOWER GARDEN, BICYCLE and CHEER.

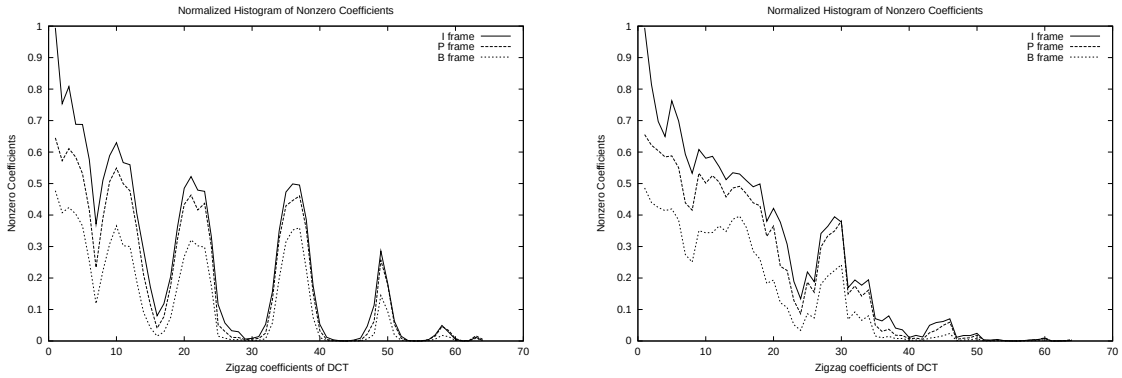
	Number of Error	SOA Err-Free	Parity Err-Free	SOA Err Blks	Parity Err Blks	SOA PSNR	Parity PSNR
FLOWER	224	19	49	2.75	2.04	22.71	23.76
BICYCLE	224	10	26	7.24	6.42	23.66	24.16
CHEER	222	19	45	3.88	3.17	25.00	26.02

6.4.3 Variable Word Lengths

In the previous section, results were presented for eight parity words of equal length. Based on these results, the possibility of a variable length word to improve the performance of the system was investigated. The results of this investigation are presented next.

As noted above, the ability of the parity bits to capture errors decreases in P frames and B frames because of fewer encoded blocks and a reduced number of non-zero coefficients in these blocks. For example, in the FLOWER GARDEN sequence, all blocks (1350 macroblocks, 4 DCT blocks for luminance and 2 DCT blocks for chromiance) are encoded in each I frame with an average of 16 non-zero coefficients. In P frames only 64.5% of the blocks are encoded with 12 non-zero coefficients and in B frames, 49.2% of the blocks are encoded with 8 non-zero coefficients. Using this knowledge, the variable word length were used as an alternative method of grouping parity bits to improve performance.

The normalized histograms of non-zero DCT coefficients for I, P and B frames of the FLOWER GARDEN sequence are presented in Figure 43. Two items of note: first is the concentration of the non-zero coefficients in the initial values, which is the energy packing feature of the DCT; and second the periodic nature of the coefficients which is caused by the interlacing of the sequence. This is especially pronounced in Figure 43(a) using the



(a) Normal Zigzag Scan. (b) Alternate Scan.

Figure 43: The histogram of non-zero DCT coefficients. **(a)** Normal zigzag scan. **(b)** Alternate scan.

normal zigzag scan and is mitigated using the alternate scan (Figure 43(b)). In an effort to exploit these two features of the histograms, more parity bits were added to the first values in the DCT and parity words were created to work in conjunction with the variation in the coefficients. The results are presented in Table 13.

The information is presented in a subjective format to limit the need to reproduce gross quantities of data. The 8 bit parity vector described in Section 6.2 is the standard for comparison. The vectors are described by the end value for each word. For example, the third entry in Table 13 groups bits 1 through 7 together, followed by bits 8 through 16, bits 17 through 29 etc. The results are baselined to the 8 bit parity vector described above with "0" representing no change, "+,-" represent slight improvements or degradation, and "++, - -" representing moderate improvements or degradation. None of the vectors significantly outperformed the standard 8 bit parity vector. As each vector was tested, the no-error PSNR was evaluated. The changes were minor and predictable with most varying less than $0.1dB$ with the maximum variation not exceeding $0.25dB$. Next, the results of the individual parity vectors will be discussed.

The second parity vector consists of 16 bit words and is presented to explore the bounds of the codec. As expected, the 16 bit vector had the smallest impact on the no-error PSNR and did not perform as well in almost all cases. The third vector attempts to

Table 13: Varying parity words and the relative improvements gained in I and B frames.

	FLOWERS			BICYCLE			CHEER		
Vector	I	P	B	I	P	B	I	P	B
8,16,24,32,40,48,56,64	Std.	Std.	Std.	Std.	Std.	Std.	Std.	Std.	Std.
16,32,48,64	- -	- -	-	-	-	-	- -	-	0
7,16,29,44,54,64	-	-	+	++	0	+	0	0	0
4,12,20,28,36,64	0	+	++	++	++	+	-	0	++
4,8,12,20,28,36,64	+	0	++	0	++	+	0	0	++
3,10,21,36,49,64	++	+	++	0	++	+	0	0	++

capitalize on the periodic variation in the non-zero coefficients noted in Figure 43(a). The results show no significant impact on the performance of the error recovery procedure. The fourth (4,12,20,28,36,64) and fifth vector (4,8,12,20,28,36,64) concentrate the parity vectors in the first elements of the zigzag scan where the maximum non-zero coefficients exist. The improvements offered by these two vectors is most noticeable in the B frame where the number of coded blocks and non-zero coefficients were at a minimum. These parity vectors have the largest impact on the no-error PSNR (albeit still minimal) since they change the greatest number of coefficients. The final vector (3,10,21,36,49,64) attempts to use the variations on the non-zero coefficients, but this time breaking the words at the maximums instead of minimums. The results are positive, but are more likely based on the concentration of the parity bits in the initial coefficients rather than the variations in the number of non-zero coefficients.

In conclusion, a case can be made for varying the parity word length for the individual frame type. A system that maintains good no-error presentation with better error recovery could use the 8 bit words for I and P frames and use the [4,8,12,20,28,36,64] vector for B frames. Such a system is trivial to implement.

6.5 Summary

This chapter presents a codec which uses steganography to enhance the resynchronization of the decoder following an error. The hidden information comes at an imperceptible decrease in picture quality while not influencing the total bit rate. In all cases, the decoder performed better than normal early resynchronization.

In closing the Paritycodec provides a marked improvement over the existing methods of error-correction and concealment while remaining compatible with the MPEG-2 standard.

CHAPTER VII

JOINT SOURCE CHANNEL ALLOCATION OF FORWARD ERROR CORRECTION

7.1 Introduction

In the previous chapters, the Steganocodec and the Paritycodec presented methods to improve error concealment and correction at the decoder. As the multimedia environment proceeds toward Internet and mobile applications, codecs must be capable of dealing with higher loss rates and significant burst lengths. Error concealment and correction is still a significant portion of error recovery in these environments, but if used in conjunction with forward error correction a more optimal solution is often available. In this chapter, an FEC allocation method that uses analysis of both the source and channel is presented. This allocation scheme is near optimal and shows significant improvement over equal error protection (EEP) and static unequal error protection methods (UEP).

7.2 Forward Error Correction in Video

As discussed in Chapter 2 classical Shannon information theory [48] states that the design of the source coder and the channel coder can be separated and the information can be transmitted error free as long as the source information can be represented by a rate below the channel capacity. The channel coder will then add FEC to the compressed stream to enable correction of transmission errors. However, such error-free transmission can only be achieved with infinite delays in implementing FEC which makes these systems impractical. Hence joint source-channel coding is an active area of research.

In general the use of FEC in a joint source-channel coding framework can be stated as follows. Let $R_C(t)$ represent the available channel capacity at time t and let $R_S(t)$ and $R_{fec}(t)$ represent the source rate and the rate of the forward error correction, respectively. Given network parameters, such as the packet loss rate (PLR) and average burst length

(ABL), find the optimal values for $R_S(t)$ and $R_{fec}(t)$ that will minimize the total distortion at the receiver under the constraint that $R_S(t) + R_{fec}(t) \leq R_C(t)$. To optimize $R_S(t)$ and $R_{fec}(t)$ will require a model for total distortion and a model for the effects of FEC on the total distortion. To this end, consider the transmission model shown in Figure 44. The video stream is input to the encoder and the resultant bitstream is packetized and Reed-Solomon (R-S) FEC is allocated to the packets. The use of Reed-Solomon and other X-OR based FEC methods provide for the recovery of lost packets in a packet group (PG). Normally a PG contains n packets of which k are information and $n-k$ are redundancy or FEC packets. With the loss of $n-k$ or fewer packets, all k information packets can be recovered. With the loss of more than $n-k$ packets, the FEC information is of no value and all undelivered information packets are lost. After transmission across a channel that can be characterized by a particular Quality of Service (QOS), the bit stream is reconstructed from the packets and lost packets are recovered to the maximum extent possible, using the FEC information. The final stage is bitstream decoding and error concealment if necessary. Using this model, this work attempts to allocate FEC so as to minimize the expected distortion at the receiver. It accomplishes this task by first providing distortion information to the packetization process by computing the spatial and temporal distortion caused by the loss of pixels and macroblocks (Section 7.3). To determine expected distortion at the receiver, one must compute the probability of packet loss, (P_L), not only based on the QOS parameters provided by the network, but also the effect of the additional protection afforded by incorporating FEC (P_{fec}) (Section 7.4). Using this information, the assignment

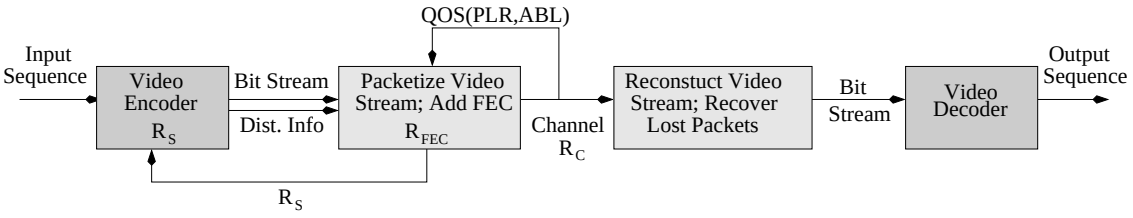


Figure 44: Video Steam is encoded and then packetized for transmission. The channel is characterized by a packet loss rate and average burst length. At the receiver, the video stream is reconstructed and losses are mitigated by FEC. The video stream is decoded and errors concealed.

of FEC can be accomplished to minimize the expected distortion at the receiver. Section 7.5 presents five alternative methods for allocating FEC. The final development uses the computed distortion information to adjust the source and FEC rates, $R_S(t)$ and $R_{fec}(t)$, to minimize the total end-to-end distortion (Section 7.6). The results of the FEC allocation and source channel rate selection are contained in Section 7.7 for both fixed rate and variable rate systems. A brief summary is included in Section 7.8.

7.3 Distortion Due to Pixel Loss

To minimize expected distortion at the receiver, a model must be developed to estimate the end-to-end distortion based on the transmission characteristics of the channel. This requires a method to determine the distortion caused by the loss of individual pixels, and, in turn, macroblocks in a block-based codec. To be more specific, this method will evaluate the effect of the loss of data in the bitstream in terms of distortion in the decoder using the following assumptions. First, the bitstream is packetized and the packet arrives either error-free or the entire packet is discarded (i.e. no bit errors). Second, if a MB spans two packets, the loss of any portion of an MB will cause the loss of the entire MB. Third, during packetization, network parameters are available to describe the packet loss rate (PLR) and the average burst length (ABL). The method is unrestricted with regard to concealment algorithms and the only restriction on the distortion measure is that it is computed on a macroblock basis.

To begin, (using notation similar to [66]), let f_n^i represent the original value of pixel i in frame n and let $\hat{f}_n^k$ represent the error free reconstruction at the encoder. Let f_{n-l}^k represent a pixel in the previous frame $n-l$ at location k that is used for motion-compensated temporal prediction. Also let $\tilde{f}_n^i$ represent the pixel at the decoder which may or may not have encountered an error. Let P_L represent the probability of loss and $P(\cdots)$ represent the probability that the event occurs; the event is the argument of the function. It is necessary to consider the different coding modes of an MB which are intra-coded, forward predicted and bidirectionally predicted. For the following discussion, it is assumed that forward temporal error concealment is used at the decoder. For the intra mode three cases

exist:

1. Error free: In this case $\tilde{f}_n^i = \hat{f}_n^i$ which occurs with the probability $(1 - P_L)$.

2. Error concealment: The lost pixel is recovered using pixel f_{n-l}^k in frame $n - l$, which was received error free. This occurs with probability $P_L P(\tilde{f}_{n-l}^k = \hat{f}_{n-l}^k)$.

3. Multiple error: The lost pixel is recovered from a concealed pixel. The probability of this event is $P_L P(\tilde{f}_{n-l}^k \neq \hat{f}_{n-l}^k)$.

For the forward predicted mode four cases exist:

1. Error free: $\tilde{f}_n^i = \hat{f}_n^i$ because of proper error-free prediction from an error-free pixel in frame $n - l$. This occurs with probability $(1 - P_L)P(\tilde{f}_{n-l}^k = \hat{f}_{n-l}^k)$.

2. Error concealment: The lost pixel is recovered using pixel f_{n-l}^k in frame $n - l$, which was received error free. This occurs with probability $P_L P(\tilde{f}_{n-l}^k = \hat{f}_{n-l}^k)$.

3. Prediction error: The current pixel is predicted from a concealed pixel. The probability of the event is $(1 - P_L)P(\tilde{f}_{n-l}^k \neq \hat{f}_{n-l}^k)$.

4. Multiple errors: The lost pixel is recovered from a concealed pixel. The probability of the event is $P_L P(\tilde{f}_{n-l}^k \neq \hat{f}_{n-l}^k)$.

For the bidirectional prediction mode five cases exits:

1. Error free: $\tilde{f}_n^i = \hat{f}_n^i$ because of proper error-free prediction from two error-free pixels in frames $n - l_1$ and $n + l_2$. This occurs with probability $(1 - P_L)P(\tilde{f}_{n-l_1}^k = \hat{f}_{n-l_1}^k)P(\tilde{f}_{n+l_2}^k = \hat{f}_{n+l_2}^k)$.

2. Forward error concealment: The lost pixel is recovered using pixel f_{n-l}^k in frame $n - l$ which was received error free. This occurs with probability $P_L P(\tilde{f}_{n-l}^k = \hat{f}_{n-l}^k)$.

3. Forward prediction error: The current pixel is recovered using a concealed pixel. The probability is $(1 - P_L)P(\tilde{f}_{n-l}^k \neq \hat{f}_{n-l}^k)P(\tilde{f}_{n+l_2}^k = \hat{f}_{n+l_2}^k)$.

4. Backward prediction error: The current pixel is recovered using a concealed pixel. The probability is $(1 - P_L)P(\tilde{f}_{n-l}^k = \hat{f}_{n-l}^k)P(\tilde{f}_{n+l_2}^k \neq \hat{f}_{n+l_2}^k)$.

5. Multiple errors: The lost pixel is recovered using a concealed pixel. The probability is $P_L P(\tilde{f}^k_{n-l} \neq \hat{f}^k_{n-l})$.

Clearly this problem is recursive in nature which, although not intuitive for humans, is well suited for computer modeling.

Additional notation is contained in Table 14 with the maximum probability of each occurrence. The term maximum probability indicates that the probability may be lower because of the fact that the pixel used for prediction, may also be predicted which indicates the recursive nature of the problem which will be developed further below.

To elaborate, the different coding types of the pixels must be considered separately. Motion-compensated temporal error concealment is the assumed method of error correction; however, the process described here is not limited by this and may be extended by referencing the appropriate pixels used for the concealment.

For an intra-coded pixel, the values of $\tilde{f}^i_n$ with its probability of occurrence are given in Equation 20

$$\tilde{f}^i_n = \begin{cases} \hat{f}^i_n & \text{with } P = (1 - P_L) \\ \bar{f}^i_n & \text{with } P = P_L(1 - P_L) \\ \ddot{f}^i_n & \text{with } P = P_L P(\tilde{f}^j_{n-l} \neq \hat{f}^j_{n-l}) \end{cases} \tag{20}$$

where

$$\bar{f}^i_n = \hat{f}^j_{n-l} \text{ for temporal concealment ,} \tag{21}$$

$$\ddot{f}^i_n = E\{\tilde{f}^j_{n-l}\} \text{ for temporal concealment ,} \tag{22}$$

and

$$\tilde{f}^j_{n-l} = 0 \text{ for } n - l < 0. \tag{23}$$

Equation 23 simply states the initial conditions. The first frame cannot be concealed because no past information exists. Others [64] [33] assume that the first frame is error free (i.e. $\tilde{f}^j_{n-l} = \hat{f}^j_{n-l}$ for $n - l = 0$)

For a forward-predicted inter-coded pixel, the value of $\tilde{f}^i_n$ with the probability of occurrence is given in Equation 24

Table 14: Notation used for description of decoded pixels.

Pixel	Description	Maximum Probability
f_n^i	Original pixel at location i in frame n.	NA
$\hat{f}_n^i$	Reconstructed pixel at encoder at location i in frame n	$(1-P_L)$
$\tilde{f}_n^i$	Decoded pixel at location i in frame n. May be correct or in error.	1
$\bar{f}_n^i$	Lost pixel correctly concealed decoder at location i in frame n.	$P_L(1-P_L)$
$\dot{f}_n^i$	Predicted pixel at location i in frame n correctly received pixel at decoder; but predicted from a concealed pixel.	$(1-P_L)P_L$
$\ddot{f}_n^i$	Lost pixel at location i in frame n concealed using an erroneous pixel for concealment.	P_L^2

$$\tilde{f}_n^i = \begin{cases} \hat{f}_n^i & \text{with } P = (1-P_L)P(\tilde{f}_{n-l}^j = \hat{f}_{n-l}^j) \\ \bar{f}_n^i & \text{with } P = P_L P(\tilde{f}_{n-l}^j = \hat{f}_{n-l}^j) \\ \dot{f}_n^i & \text{with } P = (1-P_L)P(\tilde{f}_{n-l}^j \neq \hat{f}_{n-l}^j) \\ \ddot{f}_n^i & \text{with } P = P_L P(\tilde{f}_{n-l}^j \neq \hat{f}_{n-l}^j) \end{cases} \tag{24}$$

where

$$\bar{f}_n^i = \hat{f}_{n-l}^j \text{ for temporal concealment ,} \tag{25}$$

$$\dot{f}_n^i = E\{\tilde{f}_{n-l}^j\} + e_n^i \text{ for temporal concealment ,} \tag{26}$$

$$\ddot{f}_n^= E\{\tilde{f}_{n-l}^j\} \text{ for temporal concealment ,} \tag{27}$$

and

$$\tilde{f}_{n-l}^j = 0 \text{ for } n-l < 0. \tag{28}$$

For a bidirectionally predicted inter-coded pixel, the value of $\tilde{f}_n^i$ with the probability of occurrence is given in Equation 29. Again, only forward error concealment is considered. The inclusion of bidirectional concealment is trivial.

$$\tilde{f}_n^i = \begin{cases} \hat{f}_n^i & \text{with } P = (1-P_L)P(\tilde{f}_{n-l}^j = \hat{f}_{n-l}^j)P(\tilde{f}_{n+l}^j = \hat{f}_{n+l}^j) \\ \bar{f}_n^i & \text{with } P = P_L P(\tilde{f}_{n-l}^j = \hat{f}_{n-l}^j)P(\tilde{f}_{n+l}^j = \hat{f}_{n+l}^j) \\ \dot{f}_n^i & \text{with } P = (1-P_L)P(\tilde{f}_{n-l}^j \neq \hat{f}_{n-l}^j) \text{ or } P(\tilde{f}_{n+l}^j \neq \hat{f}_{n+l}^j) \\ \ddot{f}_n^i & \text{with } P = P_L P(\tilde{f}_{n-l}^j \neq \hat{f}_{n-l}^j) \text{ or } P(\tilde{f}_{n+l}^j \neq \hat{f}_{n+l}^j) \end{cases} \tag{29}$$

where

$$\bar{f}_n^i = \hat{f}_{n-l}^j \text{ for temporal concealment ,} \tag{30}$$

$$\dot{f}_n^i = (E\{\tilde{f}_{n-l}^j\} + E\{\tilde{f}_{n+l}^j\})/2 + e_n^i \text{ for temporal concealment ,} \tag{31}$$

$$\ddot{f}_n^= E\{\tilde{f}_{n-l}^j\} \text{ for forward only temporal concealment ,} \tag{32}$$

and

$$\tilde{f}_{n-l}^j = 0 \text{ for } n-l < 0. \tag{33}$$

Now consider a distortion measure defined on a pixel basis where D_p is the distortion between the original pixel f_n^i and the received pixel $\tilde{f}_n^i$ (Equation 34)

$$D_p(n,i) = d(f_n^i, \tilde{f}_n^i) \tag{34}$$

where $d(\cdots)$ is not restricted, but is often an $L1$ or $L2$ norm. Using this distortion measure $d(f_n^i, \hat{f}_n^i)$ represents the distortion introduced as source encoding error, while $d(f_n^i, \bar{f}_n^i)$ represents the distortion introduced by error concealment and $d(f_n^i, \dot{f}_n^i)$ represents the error caused by temporal propagation. The final possible measure, $d(f_n^i, \ddot{f}_n^i)$, is the result of temporal error propagation compounded by error in the current frame.

The goal of this work is to assign FEC packets in a manner that will minimize the distortion at the receiver. To this end, the determination of the distortion which results from the loss of each pixel will drive the allocation algorithm. Since this will be applied to a block-based codec transmitted in a packet network, it is prudent to convert the pixel-level distortion to the macroblock level (and ultimately to the packet level) distortion by summing and averaging $d(f_n^i, \tilde{f}_n^i)$ over the MB of size m_0 as indicated in Equation 35

$$D_M(m,n) = 1/m_0 \sum_{i=i_0}^{i_0+m_0} d(f_l^i, \bar{f}_l^i). \tag{35}$$

To include the distortion caused by temporal propagation, $D_M(m,n)$ must be summed over all frames predicted from the corrupted frame, n (Equation 36)

$$D_M^T(m,n) = \sum_{k=n}^{l} mv(D_M(k,l)) \tag{36}$$

where $mv(\cdots)$ is a motion compensation operator and l must be selected based on the location of the loss in the group of pictures (GOP) and the frame type. Beginning with the simplest case, bidirectionally encoded frames are not used for prediction. Therefore, all errors are limited to a single frame and $k = l$. In the case of an intra-frame which may begin a GOP and is the prediction basis for the entire GOP, $k = 0$ and $l = N$ where N is the number of frames in the GOP. The predicted frame, which may be the prediction basis for all frames following has $k = n$, where n is the frame number in which the error occurred and $l = N$.

Figures 45 - 47 are the visual equivalent of the distortion measures for three frames of the TENNIS sequence encoded with an MPEG-2 codec. These figures can be viewed as each pixel's sensitivity to loss or the relative importance of each pixel in the decoding process. Figure 45 is the first I frame of the sequence (the intensity has been scaled down by a factor of 16 to enhance viewing). Because the I frame has no prior information, temporal concealment is ineffective, and the error is highly correlated with the image. The darkness around the high motion areas (indicating lower total error) results from poor motion compensation in the later frames, which, in turn, leads to intra-coded MB's or MB's coded with significant error components. Both of these coding types reduce the future frames dependence on prediction from these pixels. Figure 46 is a P frame (the intensity has been scaled down by a factor of 4). Because of the scaling factor, the total error in this frame is significantly less than for the I frame since fewer frames use this reference. The effects of significant motion are again noted here. The final image (Figure 47) is a B frame (scaled up by a factor of 16). Because the B frame errors are fixed to a single image, the error is significantly decreased; this is reinforced by the fact that the total difference in scaling between the I frame and the B frame is a factor of 256. Note that each pixel and macroblock loss is considered independently. Therefore, the inclusion of multiple errors in the same pixel representing $\ddot{f}_n^i$,

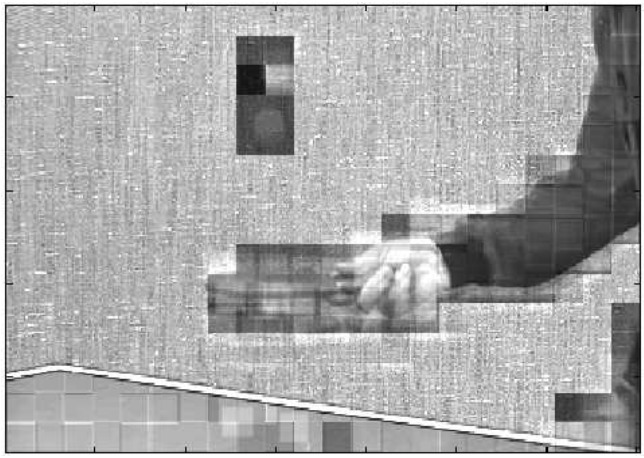


Figure 45: Pixel loss sensitivity; image is a graphical representation of the distortion measure (Equations 34-36) for an I frame in the TENNIS sequence (Scale $I/16$).

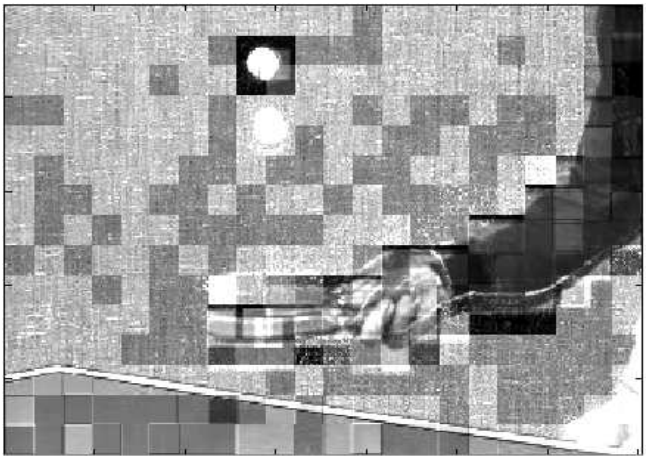


Figure 46: Pixel loss sensitivity; image is a graphical representation of the distortion measure (Equations 34-36) for an P frame in the TENNIS sequence (Scale $I/4$)

is not computed. This was excluded since it occurs with a maximum probability of P_L^2 and the desire to limit the number of possible outcomes to a manageable level.

The final result required is the distortion caused by packet loss. In a logical extension of the discussion, the loss of packet, p, will result in the loss of all MB's contained in the packet, therefore D_P (Equation 37) is the sum of the distortion from each lost MB.

$$D_P(p) = \sum_{m=mb_0}^{mb_1} D_M(m) \tag{37}$$

In Figure 48 D_P is plotted for the first 22 frames of the FLOWER GARDEN sequence using an MPEG-2 codec with six frames per group of pictures and one P frame per GOP

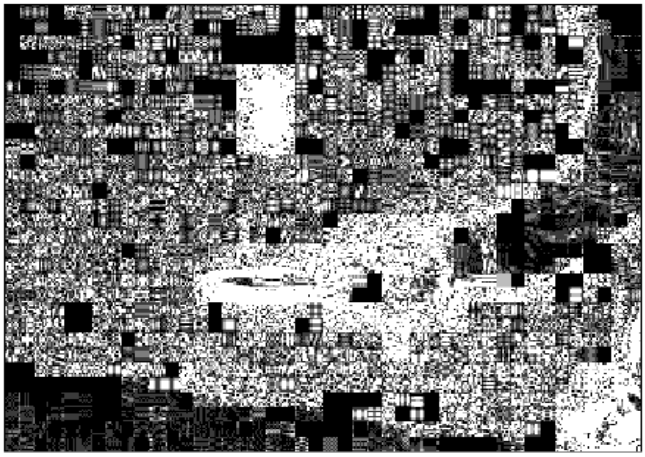


Figure 47: Pixel loss sensitivity; image is a graphical representation of the distortion measure (Equations 34-36) for an B frame in the TENNIS sequence (Scale $I \times 16$).

(i.e. $N = 6$, $M = 3$). The encoding rate is 1.8 Mbps. The stair shaped plot (dashed line) indicates the frame types as annotated. The first I frame is off scale since there is no previous data from which to apply temporal concealment. The periodic nature of the plot is the result of the very consistent panning nature of the sequence; however the nature of the picture is also suggested by the shape of the individual frames. The high peaks on the edges of the frame are indicative of the complex nature of the overhanging trees at the top and the flower garden at the bottom. The lower center portion is the easily concealed sky segment of the frame. The large variation between the distortion of the different packets indicates the difference in their relative importance to the decoder. It is this realative difference, that will be exploited to improve the allocation of FEC as presented in Section 7.5.

In this section, the distortion created by an individual packet loss is presented. To determine the expected distortion at the receiver, these equations must include a probability of loss which, in turn, is influenced by FEC. In the next section a model for packet loss with variable FEC is presented followed by the computation of the expected end-to-end distortion in Section 7.6.

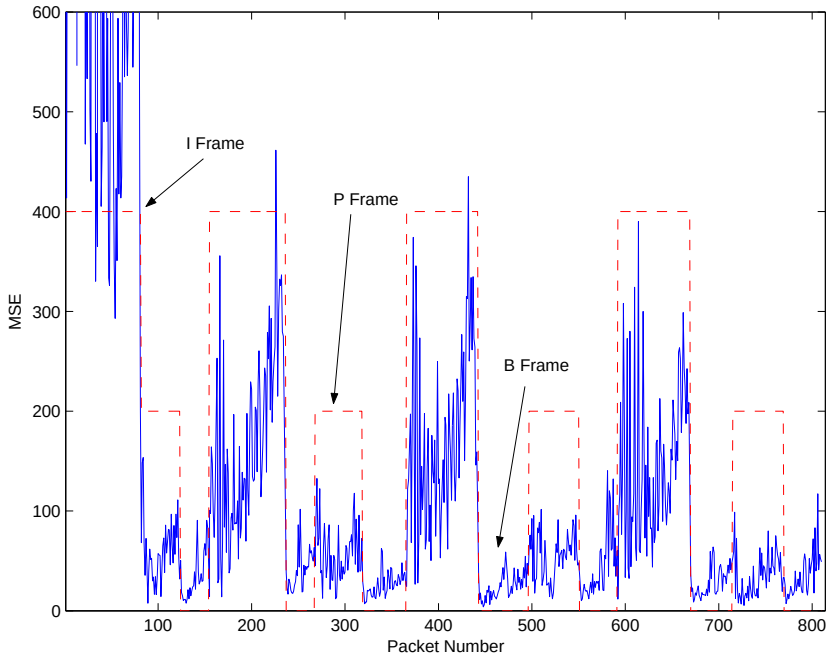


Figure 48: First 22 frames of the FLOWER GARDEN using an MPEG-2 codec with six frames per group of pictures (GOP) and one P frame per GOP (i.e. $N = 6$, $M = 3$). The encoding rate is 1.8 Mbps.

7.4 Probability of Packet Loss with Forward Error Correction

The packet loss rate (PLR) in a network is a quality of service (QOS) parameter that can be used to adjust source encoding and FEC rates; however, the use of FEC changes the packet loss rate. To predict the expected distortion at the receiver, one must use the packet loss rate seen by the receiver after accounting for FEC recovery of lost packets, (P_{fec}). This section will develop a model for predicting P_{fec} based on a bursty loss channel model.

In the current literature the Gilbert model is frequently used as an estimate of the losses on bursty communication channels [18] [17] [64]. The Gilbert model is a two-state Markov chain (Figure 49) in which G represents received packets and B represents bad or burst loss of packets. P and p represent the probability of transition from and to the G state, respectively and $Q = 1 = P$ and $q = 1 - p$ represent the probability of remaining in the receive, or loss state, respectively. The parameters of the model can be used to derive the network parameters, packet loss rate and average burst length, which will be designated P_L

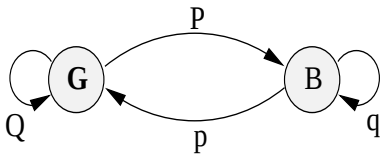


Figure 49: Transition diagram for Gilbert's Markov chain.

and α_L with $P_L = \frac{P}{p+P}$ and $\alpha = \frac{1}{p}$.

The use of the Gilbert model is complicated by the use of forward error correction. Authors [35] frequently use a Bernoulli model to predict the probability of loss with FEC (Equations 38 and 39).

$$P_{fec} = P_L(1 - P_{rec}) \tag{38}$$

$$P_{rec} = \sum_{i=0}^{n_{fec}-k_{fec}-1} \binom{n_{fec}-1}{i} P_L^i (1-P_L)^{n_{fec}-i-1} \tag{39}$$

However, the Bernoulli model assumes independent losses, and does not account for the burstiness of the Gilbert model. The comparison of the Bernoulli results with the output of the channel model is shown in Figure 50 and is unsatisfactory because it incorrectly models the FEC's effectiveness by providing overly optimistic packet loss rates.

An alternative model, albeit a more complicated one, has been developed by others [15][26][18]. The model presented here accounts for both the burstiness of the channel and the error correcting capability and limitations of the FEC. This derivation closely follows [15].

Using the notation of [15] to address the loss of packets, consider a binary model where the loss state is defined as 1 and the non-loss (receive) state is 0. The loss state will be considered a renewal error process which allows the assumption that the interval between errors is independent and uniformally distributed. Let $p(i)$ denote the probability that between errors, $i-1$ packets are received, that is $p(i) = Prob(0^{i-1}1|1)$, where 0^{i-1} denotes $i-1$ packets are received. In a similar fashion let $P(i)$ denote the probability that at least $i-1$ packets are received after an error, that is $P(i) = Prob(0^{i-1}|1)$.

Using the independence of the error intervals the events 10^{i-1} and $0^{i-1}1$ are equiprobable. Hence $R(M,n)$, the probability that $m-1$ errors occur in the next $n-1$ packets can

be computed by recurrence (Equation 40).

$$R(m,n) = \begin{cases} P(n), & m = 1 \text{ and } n \geq 1 \\ \sum_{i=1}^{n-m+1} p(i)R(m-1, n-i), & \text{for } 2 \leq m \leq n \end{cases} \tag{40}$$

Similarly, let $q(i)$ denote the probabity of $i-1$ errors and $Q(i)$ denote the probability that at least $i-1$ losses follow a received packet. With these probabilities, the dual of $R(m,n)$, $S(m,n)$ can be defined where $S(m,n)$ represents the probability of recieving $m-1$ of the next $n-1$ packets following a received packet. Again, computing by recurrence (Equation 41).

$$S(m,n) = \begin{cases} Q(n), & m = 1 \text{ and } n \geq 1 \\ \sum_{i=1}^{n-m+1} q(i)S(m-1, n-i), & \text{for } 2 \leq m \leq n \end{cases} \tag{41}$$

Equations 40 and 41 can be used to determine the packet loss rate with FEC. Using the properties of the renewal process, if the last packet of the last PG was lost, then the packet loss rate, P_L^1 is

$$P_L^1 = \frac{1}{k} \sum_{i=1}^{k} iR(i,k) \sum_{j=\lfloor n-k+1-i \rfloor}^{n-k} R(j+1, n-k+1). \tag{42}$$

Where the $\frac{1}{k}$ accounts for the fact that only one of the k possible loss events occurs, while the first summation accounts for the number of lost information packets and the second summation accounts for the loss of FEC packets with $\lfloor x \rfloor$ denoting $x : x \geq 0$.

In a similar sense, if the last packet of the last PG was received then the packet loss rate, P_L^0 is

$$P_L^0 = \frac{1}{k} \sum_{i=1}^{k-1} (k-i)S(i,k) \sum_{j=0}^{k-1-i} S(j+1, n-k+1). \tag{43}$$

Again, the $\frac{1}{k}$ accounts for the fact that only one of the k possible loss events occurs; the first summation accounts for the number of lost information packets, and the second summation accounts for the loss of FEC packets. Assuming the loss of the last packet is random, the global packet loss rate, P_{fec} is

$$P_{fec} = P_L P_L^1 + (1 - P_L) P_L^0. \tag{44}$$

Combining Equations 42 - 44 the global packet loss rate is

$$\begin{aligned} P_{fec} = \frac{P_L}{k} \sum_{i=1}^{k} iR(i,k) \sum_{j=\lfloor n-k+1-i \rfloor}^{n-k} R(j+1, n-k+1) \\ + \frac{(1-P_L)}{k} \sum_{i=1}^{k-1} (k-i) S(i,k) \sum_{j=0}^{k-1-i} S(j+1, n-k+1). \end{aligned} \tag{45}$$

The only factors remaining are the definitions of $p(i)$, $P(i)$, $q(i)$, and $Q(i)$ which are easily determined for the Gilbert model by inspection (Equations 46 - 49).

$$p(i) = \begin{cases} 1-q, & \text{if } i = 1 \\ q(1-p)^{i-2}p, & \text{otherwise} \end{cases} \tag{46}$$

$$P(i) = \begin{cases} 1, & \text{if } i = 1 \\ q(1-p)^{i-2}, & \text{otherwise} \end{cases} \tag{47}$$

$$q(i) = \begin{cases} 1-p, & \text{if } i = 1 \\ p(1-q)^{i-2}q, & \text{otherwise} \end{cases} \tag{48}$$

$$Q(i) = \begin{cases} 1, & \text{if } i = 1 \\ p(1-q)^{i-2}, & \text{otherwise} \end{cases} \tag{49}$$

Figure 50 shows the modeling results for four possible channel states ($P_L = 1\%$,10% and $\alpha = 4$,8). Packet loss rates are plotted as a function of k with $n = 20$ with graphs depicting the model and discrete points representing the results of Gilbert model simulations. The Binomial model is included, but the results are unsatisfactory because of its overly optimistic packet loss rate.

The model for packet loss in conjuntion with FEC agrees well with data collected using the Gilbert model. In the next section, the expected loss combined with the distortion due to packet loss will be combined to yield the total end-to-end expected distortion due to packet loss.

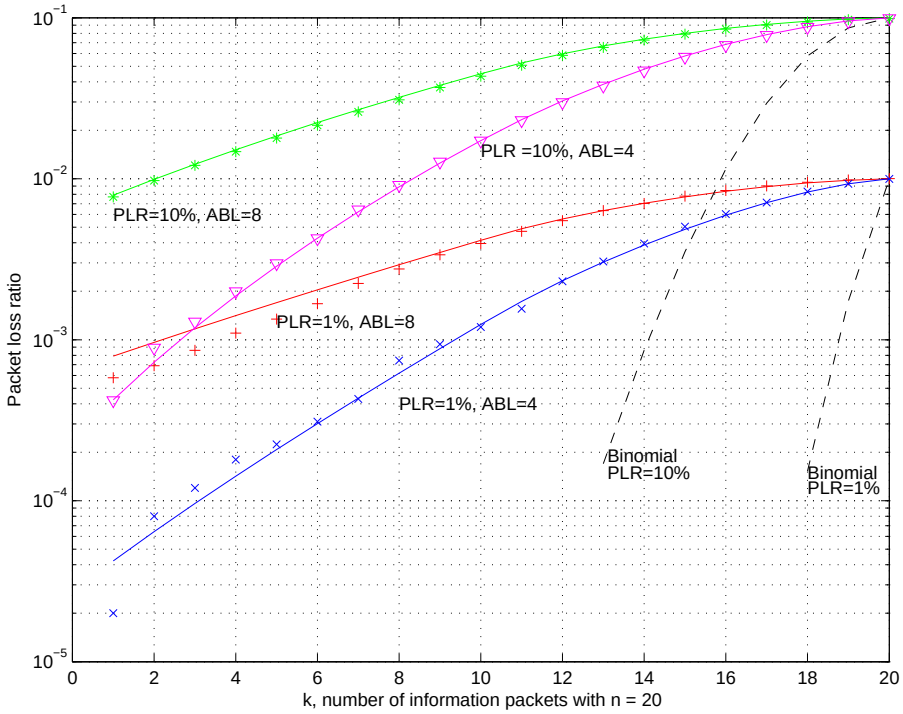


Figure 50: Packet loss rates as a function of k with $n = 20$. Graphs depict model and discrete points are results of gilbert model simulations. The Binomial model (dashed) is included but results are unsatisfactory because of its overly optimistic packet loss rate.

7.5 FEC Allocation

The allocation of FEC in packetized data presents a challenging problem and in this section the first portion of the joint source-channel coding problem will be addressed. Under the constraint that $R_S(t) + R_{fec}(t) \leq R_C(t)$, the following discussion will assume that R_S and R_{fec} are fixed and FEC must be allocated among the packets in a packet group (PG). Two existing methods are Equal Error Protection (EEP) and Unequal (albeit fixed) Error Protection (UEP). In an effort to improve upon these allocation schemes, this work presents an FEC allocation scheme that minimizes the expected distortion for the packet group (PG).

To begin, a group of T total packets (TG) is divided into $N = T/k$ packet groups (PG) and the amount of FEC to allocate is $N \times (n - k)$ (Figure 51). The expected distortion due to packet loss for packet p, can be estimated since the distortion is caused by the loss (Equation 37) and the probability of loss is known.

$$E\{D_P(p)\} = P_{fec}(p)D_P(p) \tag{50}$$

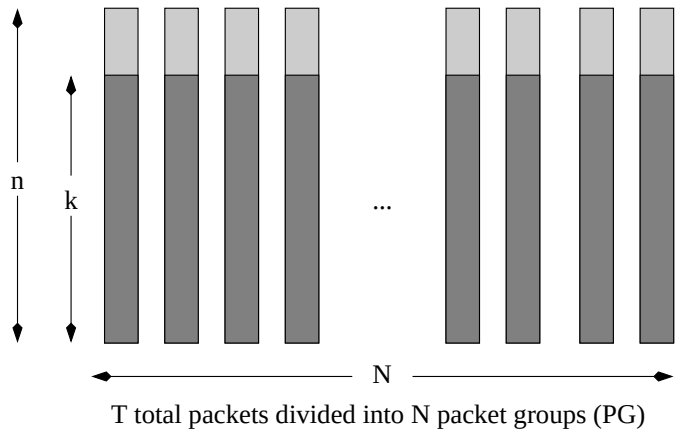


Figure 51: A group of T total packets (TG) is divided into $N = T/k$ packet groups (PG) and the amount of FEC to allocate is $N \times (n - k)$.

where P_{fec} for each packet p is based on the amount of FEC alloted to packet p. The expected distortion of the total packet group is

$$E\{D_{TG}\} = \sum_{i=0}^{T-1} P_{fec}(p) E\{D_P(p)\}. \tag{51}$$

This problem resembles the allocation of FEC packets in set partitioning in hierarchical trees (SPIHT) [27] [38] with two exceptions. First, because of the hierarchical partitioning of the data with SPIHT, packet $i+1$ contains less information than packet i for all packets and the loss of packet i will cause the loss of all remaining packets (i.e. $i + 1...N - 1$); however the problem is still insightful since both problems concern the allocation of FEC to a group of packets of varying value in an effort to minimize loss distortion. In [38] all packets are initialized with equal error protection. The loss due to each packet, $D_P(i)$, is computed and an FEC packet is removed from the $D_P(min)$ and placed in $D_P(max)$. The algorithm continues until a convergence criteria is reached. In this work this method is referred to as the minimum/maximum (MM) method.

Computational requirements for this method are determined by the number of iterations, I and the distance between the minimums and maximums which can be approximated by $\frac{N}{2}$. For each iteration, the D_P for two packets is recomputed based on a new (n, k) which requires a subtraction or addition and two multiplications. Each packet between the min and max require the shifting of one packet which requires one addition, one subtraction and

a multiplication. Hence the computation requirements are

$$Comp = I\frac{N}{2}(adds) + I\frac{N}{2}(subs) + I(\frac{N}{2}+1)(mults). \tag{52}$$

The alternative method proposed here is the optimal allocation (OPT) of each packet based on minimizing the distortion of the entire packet group D_{TG}. To begin, for each packet $n = k$, i.e. no FEC, and the following algorithm is continued until all available FEC is allocated.

1. Compute $D_{TG}(i)$ with $i : 0...N-1$ where an FEC packet is added to $PG(i)$
2. Add FEC to $PG(i)$ so as to minimumize D_{TG}.
3. Continue until all FEC is allocated.

Using this algorithm each FEC packet is optimally allocated to minimize $D_{PG}(i)$ in each interation. The computational requirements for this method are as follows. For each iteration, $N(n-k)$, D_{PG} must be computed based on adding FEC and a shift of packets (FEC added in a previous packet). The computation of a shift is one add, subract, and multiply. The computation of an add FEC is one subtract and one multiply. $D_{TG}(i)$ requires one additional subtract and add based on the value of $D_{TG}(i+1)$ in which $D_{TG}(i)$ is computed from $N-1\cdots 0$ and $D_{TG}(N) = D_{TG}(\text{Last iteration})$. The algorithm continues until all FEC has been allocated; therefore the computation requirements are

$$\begin{aligned} Comp &= N(n-k)(N(add+sub+mult)+\cdots \\ &\qquad N(sub+mult)+N(add+sub) \\ &= N(n-k)(5N(add/sub)+2N(mult)) \end{aligned} \tag{53}$$

A heuristic alternative to this method with considerably lower computational requirements is the Maximum (MAX) algorithm which begins as the OPT method with no FEC allocated and adds 1 packet of FEC to the $PG(i)$ with the highest distortion. The computational requirments for this are the allocation of $N(n-k)$ packets each requiring an average of $N/2(add+sub+mult)+N(adds)$.

The final two algorithms are static algorithms that require no significant computational requirements. The first is equal error protection in which each packet is provided $n-k$ packets of FEC. The second is UEP which applies approximately 40% FEC protection for intra coded frames, 20% FEC for predicted frames and 5% FEC for bidirectionally predicted frames.

The three non-fixed methods to allocate FEC to packet groups in this section attempt to minimize the distortion caused by losses under the assumption that each packet loss is independent. To determine a total end-to-end distortion estimate, the assumption of independence must be removed, which is the topic of the next section.

7.6 End-to-End Expected Distortion

7.6.1 Expected Distortion

To this point packet loss and the expected distortion is computed under the assumption that each packet loss is independent and that the source encoding rate, R_S, and the FEC rate, R_{fec}, are fixed. These assumptions provide an adequate framework for the allocation of allotted FEC; however, to further this work one would like to evaluate FEC requirements in a complete Rate-Distortion framework in which $E\{D_T\}$ is minimized. This requires the judicious choice of R_S and R_{fec} under the constraint that $R_S(t) + R_{fec}(t) \leq R_C(t)$ and requires an estimate of the total expected end-to-end distortion. To begin, the expected value for f_n^i in an intra-coded pixel is

$$E\{\tilde{f}_n^i\} = \hat{f}_n^i(1-P_L) + \bar{f}_n^i P_L(1-P_L) + \ddot{f}_n^i P_L^2 \tag{54}$$

Similarly, for a forward predicted pixel, the expected value is

$$\begin{aligned} E\{\tilde{f}_n^i\} &= \hat{f}_n^i(1-P_L)P(\tilde{f}_{n-l}^j = \hat{f}_{n-l}^j) + \bar{f}_n^i P_L P(\tilde{f}_{n-l}^j = \hat{f}_{n-l}^j) + \\ &\quad + \dot{f}_n^i(1-P_L)P(\tilde{f}_{n-l}^j \neq \hat{f}_{n-l}^j) + \ddot{f}_n^i P_L P(\tilde{f}_{n-l}^j \neq \hat{f}_{n-l}^j) \end{aligned} \tag{55}$$

And for a bidirectionally coded pixel

$$\begin{aligned} E\{\tilde{f}_n^i\} &= \hat{f}_n^i(1-P_L)P(\tilde{f}_{n-l}^j = \hat{f}_{n-l}^j)P(\tilde{f}_{n+l}^j = \hat{f}_{n+l}^j) + \\ &+\bar{f}_n^i P_L P(\tilde{f}_{n-l}^j = \hat{f}_{n-l}^j)P(\tilde{f}_{n+l}^j = \hat{f}_{n+l}^j) + \\ &+\dot{f}_n^i(1-P_L)P(\tilde{f}_{n-l}^j \neq \hat{f}_{n-l}^j)P(\tilde{f}_{n+l}^j = \hat{f}_{n+l}^j) + \\ &+\dot{f}_n^i(1-P_L)P(\tilde{f}_{n-l}^j = \hat{f}_{n-l}^j)P(\tilde{f}_{n+l}^j \neq \hat{f}_{n+l}^j) + \\ &+\ddot{f}_n^i P_L P(\tilde{f}_{n-l}^j \neq \hat{f}_{n-l}^j) \end{aligned} \tag{56}$$

Using the expected values, the expected distortortion is $E\{D_p(n,i)\} = d(f_n^i, E\{\tilde{f}_n^i\})$. Unfortunately, since most distortion measures are non-linear (e.g. mean absolute difference) the computation of $E\{D_p\}$ quickly becomes intractable. In an effort to estimate $E\{D_p\}$ consider the following. The source encoding distortion D_S, ($\tilde{f}_n^i = \hat{f}_n^i$), occurs with a maximum probability of $1 - P_L$. This value is computed by the encoder, but is only available for the current R_S; hence it must be estimated for varying bit rates. This will be discussed below. The distortion due to loss and concealment, D_l, to include propagation ($\tilde{f}_n^i = \bar{f}_n^i$ and $\tilde{f}_n^i = \dot{f}_n^i$) is also computed and occurs with a maximum probability of $P_L(1-P_L)$. Finally, multiple errors ($\tilde{f}_n^i = \ddot{f}_n^i$) occur with a maximum probability of P_L^2 and are not computed.

Thus, an estimate of the total expected distortion is (Equation 57)

$$D_T = D_S(1-P_L) + D_l P_L(1-P_L) \tag{57}$$

In a constant bit rate framework, the minimization of D_T by allocating the bandwidth between source encoding and error protection is the goal of the joint source-channel coding algorithm. This allocation requires an estimate of the source encoding error D_S for varying bit rate. A method to estimate D_S is presented next.

7.6.2 Estimating Source Encoding Error

In the previous section, it was shown that the expected distortion could be used to allocate FEC packets in a more efficient manner using the Quality of Service metrics of packet loss rate (PLR) and average burst length (ABL). The results indicate that a best choice in allocating FEC can provide the best expected distortion by balancing error losses against

source coding distortion in a constant bit rate framework. In this section, these results will be extended to provide a variable FEC allocation capable of adjusting to changing network parameters.

The total expected distortion, D_T, from Equation 57, is a combination of source encoding distortion, D_S and distortion due to packet loss, D_L. Models for D_L and P_L have been presented, and now the model for D_S will be presented.

Numerous studies have been conducted to determine the rate-distortion characteristics of MPEG coders [34][13] [11]. Exponential models have been proposed [19] and although not ideal over large rate fluctuations [34] the model works satisfactorily in this application. Using MSE-based metrics the distortion increases exponentially with decreasing source rate (Equation 58)

$$D_S(R_S) = \chi_S R_S^{\epsilon_S}. \tag{58}$$

where R_S is the source rate, ϵ_S relates to the encoding complexity of the video sequence and χ_S is a constant used for curve fitting. The selection of ϵ_S and χ_S can be accomplished off line, determined dynamically, or selected from a list of test values. In this work, these values were determined off line and are presented in Figure 52 for the FLOWER GARDEN, FOOTBALL and TABLE TENNIS sequences.

7.6.3 Rate Selection

In the previous two sections, estimated values for $D_S(R_S)$ and $D_L(R_{fec})$ have been presented. At the end of each group of pictures, the average D_L is computed and $P_L(R_{fec})$ is computed over the range $R_{fec}/R_C \in [5\%, 50\%]$ and the total expected distortion is estimated as

$$D_T(R_S) = D_S(R_S) + P_L(\frac{R_C - R_S}{R_C})D_L. \tag{59}$$

The minimum value is chosen, the FEC is allocated to the packets to minimize distortion in the PG, and the encoding rate is adjusted as appropriate. This method assumes an average loss induced distortion D_L per frame and assumes that the FEC is allocated to limit P_L uniformly across the frames as is the case in an EEP system. Although this is

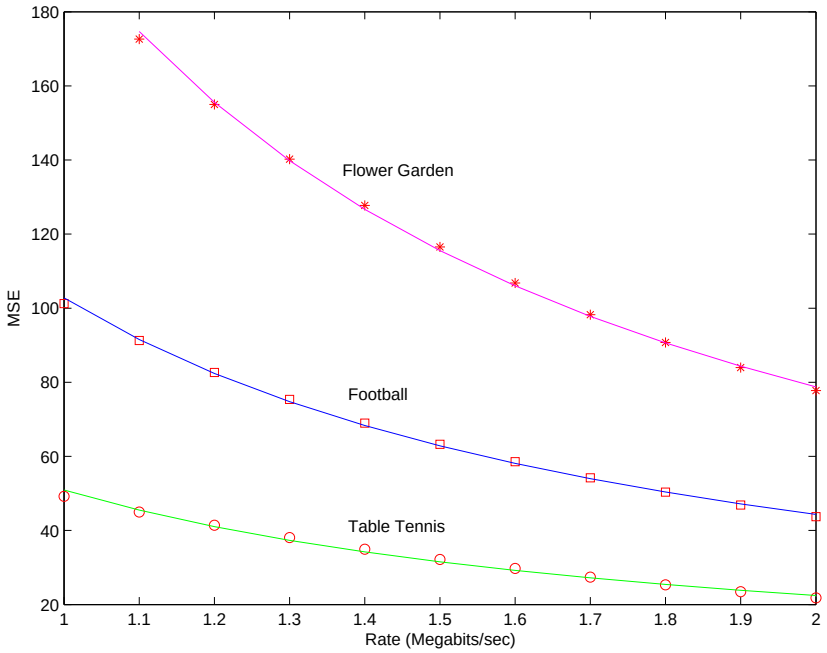


Figure 52: Rate-Distortion for the FLOWER GARDEN, FOOTBALL and TABLE TENNIS sequences. Discrete points are codec results; curves represent the modeled values.

a limitation, the alternative requires an interactive process in which $D_L P_L$ is recomputed each time by running the allocation algorithm, which would increase the computational complexity linearly with the number of rates estimated. In the next section, it will be shown that this is not a serious limitation.

In this and the previous sections, the spatial and temporal distortion caused by a packet loss was evaluated and estimates for packet loss, expected distortion due to losses, and source encoding error have been presented. In the next section, results for the allocation algorithms in both fixed and variable rate systems will be presented.

7.7 Results

7.7.1 Fixed Rate

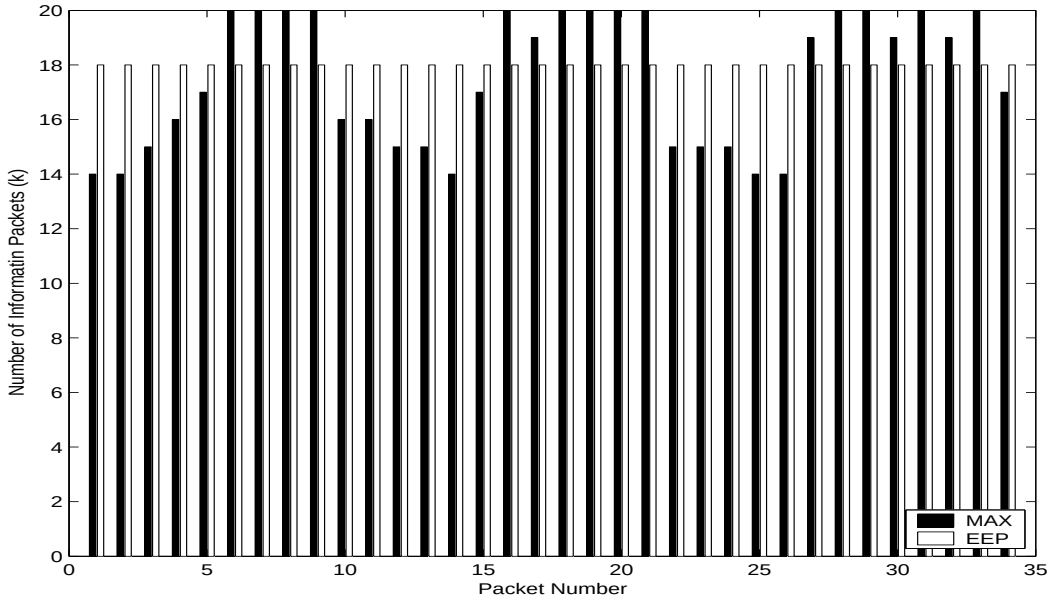
Before the results for five FEC allocation algorithms are presented, a brief overview of the method of data collection is necessary. For each allocation method, the bitstream is encoded using an MPEG-2 compliant codec with R_C equal to 2.0 Mbps, and packetized with FEC allocated in each of five methods: equal error protection (EEP); unequal error

protection (UEP); optimal method (OPT); minimum/maximum (MM) method; and maximum (MAX) method. Each bitstream is transmitted via a lossy network simulated by a Markov loss model for four to ten repetitions with the final results averaged. Errors in the transmitted sequences are recovered using temporal concealment. It is assumed that concealment vectors are derived from the nearest six neighbors (above and below) and resynchronization information is available. The parameters varied as follows: packet loss rate, $PLR \in [.01, .15]$; average burst length, $ABL \in [1, 8]$; and forward error correction, $FEC \in [5\%, 50\%]$. This matrix of parameters generated a large amount of data. Since the data was well behaved, in an effort to reduce clutter, the results are presented near each bound.

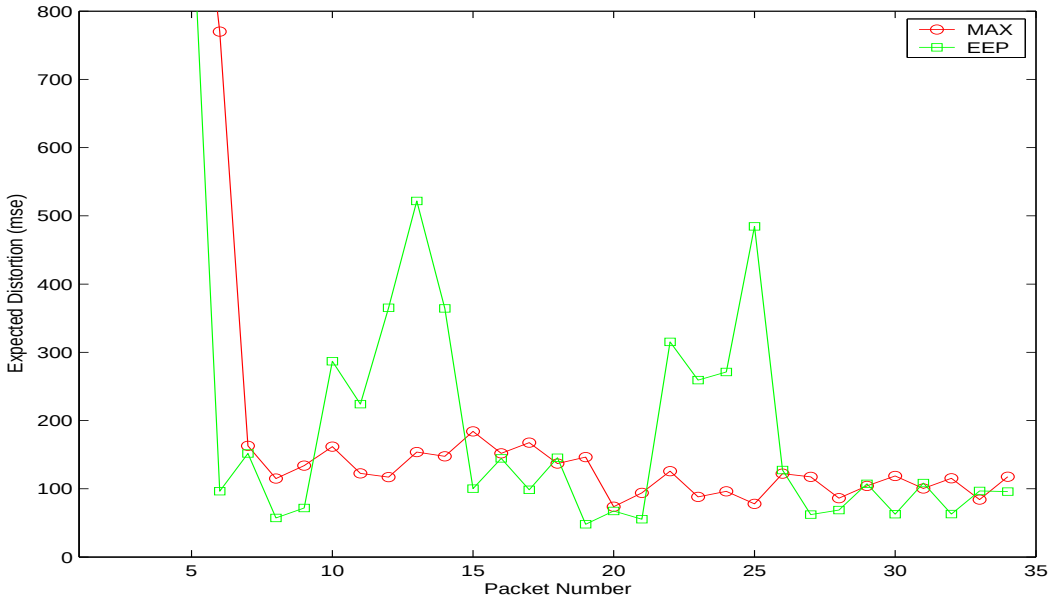
Results for the UEP scheme were not acceptable. The UEP scheme uses a different quantity of FEC to protect different frame types. The intra coded frames received 40% FEC, forward predicted frames 20% and bidirectional frames 5%. With the encoding rates used in this study, the average FEC allocation was approximately 32%. In all cases, the UEP scheme underperformed all other methods of allocation and will not be considered further.

The primary purpose of the allocation algorithms is to add additional protection to packets groups (PG) with the highest expected distortion. The first results presented in Figure 53 contrast the EEP allocation algorithm with the MAX algorithm using the first 16 frames of the FLOWER GARDEN sequence with $n = 20$, $k = 18$, 1.8 Mbps, $PLR = 0.10$ and $ABL = 2$. Figure 53(a) is a bar plot of the values of k for the packet groups with the Group of Pictures (GOP) beginning at packet numbers 1, 9 and 22. The EEP algorithm maintains a constant $k = 18$, while the MAX algorithm varies k to minimize expected distortion. Figure 53(b) plots the expected distortion for the PG's. The first several packets are off scale because they represent the distortion for the first frame which has no temporal error concealment information available. From the remainder of the plot, it is clear that the MAX algorithm equalizes the expected distortion and, as will be shown next, produces a lower total distortion at the receiver.

The next results are presented for the FLOWER GARDEN sequence, which is spatially

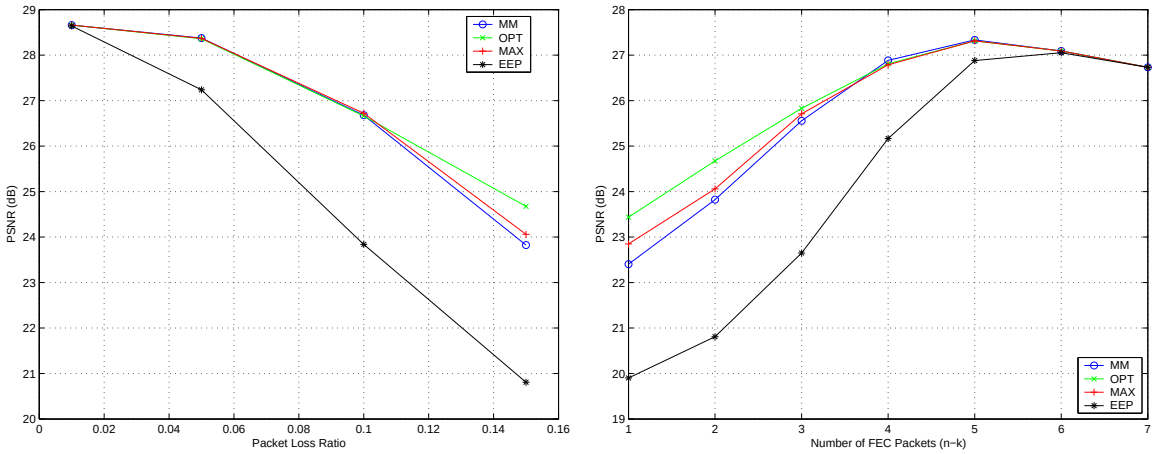


(a)



(b)

Figure 53: EEP and MAX FEC allocation algorithm using the first 16 frames of the FLOWER GARDEN sequence with $n = 20$, $k = 18$, 1.8 Mbps, $PLR = 0.10$ and $ABL = 2$. GOP begins at packet numbers 1, 9 and 22. **(a)** Bars indicate value of k. **(d)** Expected distortion.

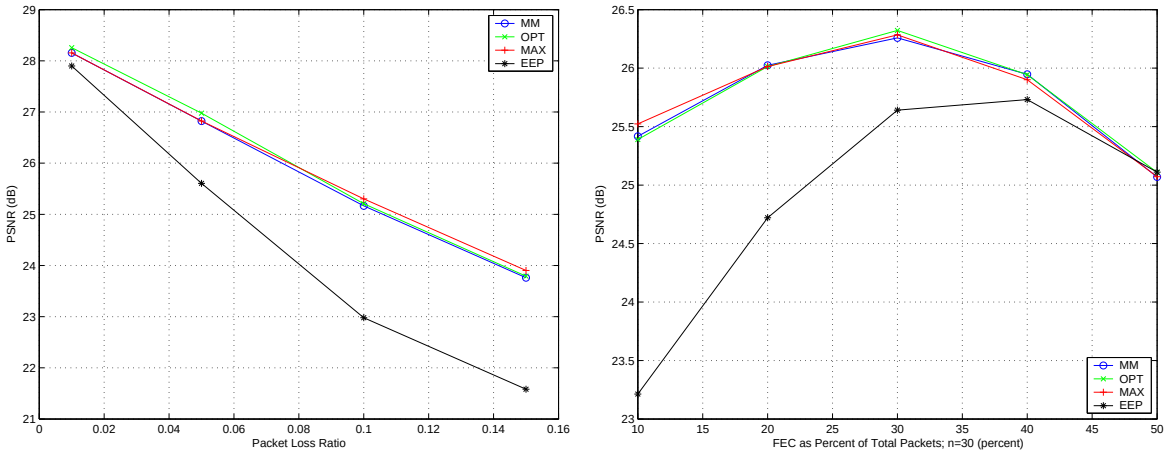


(a). (b).

Figure 54: FLOWER GARDEN: **(a)** PSNR vs. PLR with with $ABL = 1$, $n = 20$ and $k = 18$. **(b)** PSNR vs. FEC with $PLR = 0.15$, $ABL = 1$, and $n = 20$.

complex, but contains simple panning motion. The first results presented plot PSNR vs packet loss rate as presented in Figure 54(a) with $ABL = 1$, $N = 20$ and $k = 18$. Several trends can be noted in this figure. When the FEC can adequately compensate for the errors (left side of the graph) all the allocation schemes perform well and the total distortion is defined primarily by the source distortion. When the FEC cannot compensate for all the error, the distortion-based allocation schemes perform much better than equal error protection, and finally when the packet loss rate is high, the optimal scheme performs better than the minimum/maximum and maximum based allocation schemes. Similar trends are present in Figure 54(b) which plots PSNR vs. FEC with $N = 20$ with a $PLR = .15$ and $ABL = 1$. In addition, this plot demonstrates that there is a combination of FEC and source coding that will maximize the received PSNR as expected since inadequate FEC leads to errors from transmission losses and excessive FEC will increase the source coding error due to the constant bit rate constraint.

Figures 55 thru 56 show similar results for increased burst lengths. To improve the performance of the FEC, the size of the packet group was increased to $N = 30$. Note the results are similar, but with some important points of note. As in the single packet loss data, there exists a optimal point for FEC allocation based on the bit rate and packet loss



(a). (b).

Figure 55: FLOWER GARDEN: **(a)** PSNR vs. PLR with $ABL = 4$, $n = 30$ and $k = 27$. **(b)** PSNR vs. FEC with $PLR = .10$, $ABL = 4$ and $n = 30$.

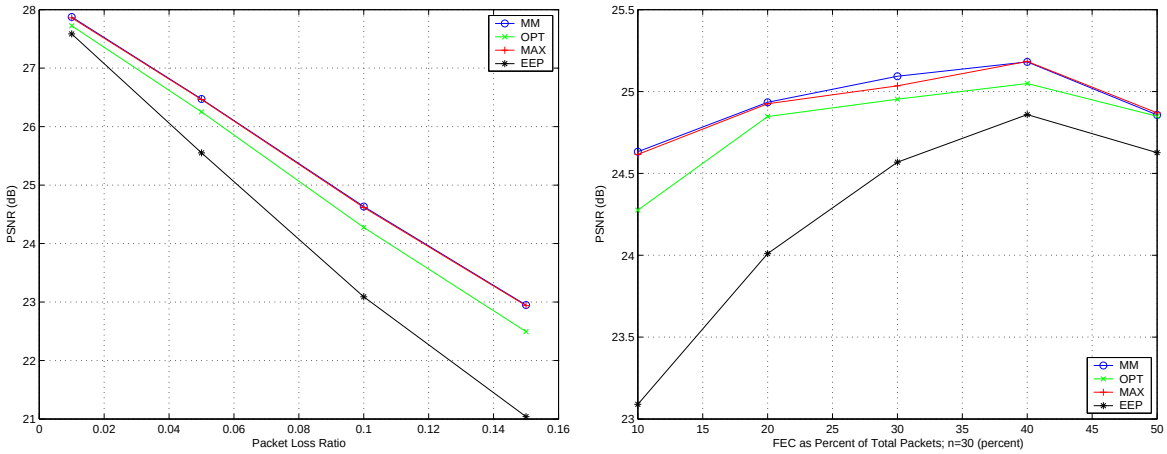
rate. Also, the improvement offered by the use of the FEC greatly decreases with increased burst length. For example in the case of a $PLR = 0.01$, in the single packet loss experiments the improvement observed between no FEC and best is over $6dB$ ($21.41dB$ to $27.80dB$), while the maximum improvement with a burst length of 8 was less than $2.5dB$ ($22.82dB$ to $25.18dB$).

The next item of interest is the performance of the OPT method. In the single packet loss trial, the OPT method was capable of providing improved performance in cases where the FEC was insufficient. However, the OPT method shows no performance improvement when the $ABL = 4$ and, in fact, shows a degraded performance with $ABL = 8$. The cause of this will be discussed next.

Trace data indicates that improvement offered by additional FEC packets it not a monotonically decreasing function; hence, the allocation algorithms may become trapped in a local minimum. The reason for this can be seen by examining Equation 51 (repeated here) and Figure 57.

$$E\{D_{TG}\} = \sum_{i=0}^{T-1} P_{fec}(p) E\{D_P(p)\}.$$

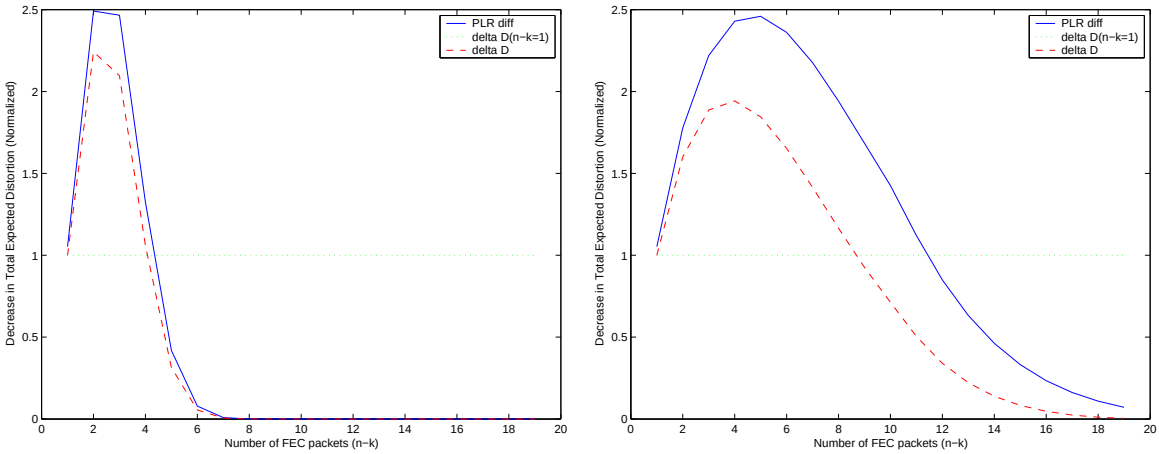
The distortion generated by a given packet is largely governed by frame type with I frames the largest, P frames next and B frames the smallest; hence the distortion of adjacent



(a). (b).

Figure 56: FLOWER GARDEN: **(a)** PSNR vs. PLR with $ABL = 8$,$n = 30$ and $k = 27$. **(b)** PSNR vs. FEC with $PLR = .10$, $ABL = 4$ and $n = 30$..

packets and in turn adjacent PG's is similar. Now consider the addition of an FEC packet to a PG: the improvement (reduction in the probability of loss) offered by an additional FEC packet is not constant as shown in Figure 57 (for $ABL = 4$ and $ABL = 1$) and labeled *PLR diff* $(P_L(R_{fec} = (n-k)/n) - P_L(R_{fec} = (n-k-1)/n)$. This non-linear reduction results in the algorithm finding non-optimal solutions as follows. The first FEC packet is added to the PG with the highest distortion which yields the greatest reduction in total distortion. This reduction in total expected distortion is the result of the decreased probability of loss but is mitigated by the reduced number packets in the PG. The normalized reduction realized by adding the first FEC packet to a PG is represented by the constant in Figure 57 which is $0.95 * P_L(R_{fec} = 5\%)$ (Using $n = 20$ and $k = 19$) and labeled *delta D(n-k=1)*. As additional FEC packets are added to the same PG, the total expected distortion is further decreased by the decrease in probability of loss; but is mitigated by the decrease in the number of packets in the PG. This is shown in Figure 57 by the line labeled *delta D* which is $((n-k)/n) * P_L(R_{fec} = (n-k)/n)$. The FEC allocation algorithm will continue to allocate FEC to the same PG until the decrease in total expected distortion is less than the decrease realized by adding the first FEC packet to another PG. This point is the intersection of the constant line at approximately $n - k = 8$ for an $ABL = 4$ and $n - k = 4$ for an $ABL = 1$.



(a) ABL = 1.

(b) ABL = 4.

Figure 57: Change in Total expected distortion vs. number of FEC packets $(n-k)$ **(a)** ABL of 4 and **(b)** ABL of 1.

This example radically simplifies the randomness of actual expected distortions, but trace data (for $ABL = 4$) confirms that the algorithm consistently added FEC to a single PG until it had added between 7 and 9 packets and then continued the same pattern on the PG with the next largest expected distortion. Although this effect is still present in shorter burst length trials, the algorithm was able to break out of the local minimum sooner. The addition of FEC can be accomplished in a number of methods; however, the proof that this is not the optimal solution lies in the fact the other non-optimal algorithms outperformed the OPT solution in cases with longer average burst length.

The results for the minimum-maximum (MM) algorithm very closely match those of the maximum algorithm (MAX). The results of these methods were consistently among the best; the choice between the two can be based on implementation. If a fixed amount of FEC is allocated the MM algorithm may be advantageous due to the limited computational requirements since convergence was normally complete in less than 25 iterations; however the MAX algorithm can dynamically add FEC packets until a threshold in expected distortion is reached.

The algorithms presented here compute the expected distortion based on network parameters; hence the addition of variable FEC based on network QOS and expected source-channel distortion is a logical extension and will be presented next.

7.7.2 Variable Rate

In the previous section, it was shown that the expected distortion can be used to allocate FEC packets in a more efficient manner using the Quality of Service (QOS) metrics of packet loss rate (PLR) and average burst length (ABL). The results indicate that a best choice in allocating FEC can provide the best expected distortion by balancing error losses against source coding distortion in a constant bit rate framework. In this section, these results will be extended to provide variable FEC allocation capable of adjusting to changing network parameters (Tables 15 through 18).

In the variable FEC algorithm, the total expected distortion, D_T, is computed based on D_L and the estimated P_L for differing quantities of FEC over a range [0%,50%] and the minimum is selected. The FEC is allocated and the bit rate is adjusted to maintain a constant bit rate. The results for the FLOWER GARDEN sequence using an EEP FEC allocation method are shown in Table 15.

The distortion metric used is PSNR and the rate associated with each is Megabits/sec. (Mbps) for the source encoding with the total bit rate with FEC a constant 2.0 Mbps. The first is the packet loss rate and the next two columns show the best results for EEP based on fixed bit rates from the last section. Columns four and five show the results for the variable-rate addition of FEC using equal error protection (EEP). The results show excellent agreement indicating that the variable rate algorithm was capable of finding the best source rate for the allocation of FEC using EEP.

Next consider the variable rate FEC allocation using the MM methods (Table 16). Again, the first column is the PLR and the next two colums show the best performance for the MM algorithm using the fixed rate allocation scheme from the last section. The next two colums show the performance and rate for the variable rate scheme. Although the MM results are presented the difference between the algorithms (MM, OPT, MAX)

Table 15: Results for the Best Fixed Rate encoding contrasted with Variable Rate using EEP for the FLOWER GARDEN sequence. Rate is the source coding rate with FEC added for a constant bit rate of 2.0 Mbps.

PLR	Best EEP (dB)	Rate EEP (Mbps)	Var EEP (dB)	Var Rate (Mbps)
0.01	27.94	1.70	27.95	1.76
0.05	26.95	1.50	26.86	1.47
0.10	26.40	1.40	26.33	1.30
0.15	25.79	1.20	25.79	1.21

Table 16: Results for the Best Fixed Rate encoding contrasted with Variable Rate using Minimum/Maximum allocation method for the FLOWER GARDEN sequence. Rate is the source coding rate with FEC added for a constant bit rate of 2.0 Mbps.

PLR	Best MM (dB)	Best Rate (Mbps)	Var MM (dB)	Var Rate (Mbps)
0.01	28.37	1.80	28.33	1.76
0.05	27.37	1.70	27.10	1.47
0.10	26.60	1.40	26.38	1.30
0.15	26.06	1.30	25.95	1.21

was typically less than $0.05dB$. The variable bit rate FEC algorithm does not select the best rate; however, this is expected since D_T is computed using an estimated P_L based on EEP. As noted in the previous section the alternate FEC allocation schemes provided the best results with one less packet of FEC. Based on this observation, a heuristic method was used to improve the results by setting $k = k + 1$. The results for the FLOWER GARDEN and TABLE TENNIS sequences are shown in Table 17 and 18 respectively.

The results from the FLOWER GARDEN show the algorithm is capable of selecting the correct FEC allocation for the current conditions but offers no improvement over the fixed

Table 17: Results for the Best Fixed Rate encoding contrasted with improved Variable Rate selection using Minimum-Maximum allocation method for the FLOWER GARDEN sequence. Rate is the source coding rate with FEC added for a constant bit rate of 2.0 Mbps.

PLR	Best Fixed (dB)	Fixed Rate (Mbps)	Var MM (dB)	Var Rate (Mbps)
0.01	28.37	1.80	28.42	1.88
0.05	27.37	1.70	27.35	1.58
0.10	26.60	1.40	26.55	1.39
0.15	26.10	1.30	26.09	1.29

Table 18: Results for the Best Fixed Rate encoding contrasted with improved Variable Rate selection using Minimum/Maximum allocation method for the TABLE TENNIS sequence. Rate is the source coding rate with FEC added for a constant bit rate of 2.0 Mbps.

PLR	Fixed Best (dB)	Fixed Rate (Mbps)	Var MM (dB)	Var Rate (Mbps)
0.01	35.32	1.90	35.71	1.84
0.05	34.35	1.60	34.56	1.56
0.10	33.51	1.40	33.87	1.39
0.15	33.02	1.30	33.21	1.26

allocation levels; however, in the TABLE TENNIS sequence, the variable allocations offers as much as a $0.40dB$ improvement. The reason for this improvement is demonstrated in Figures 58 and 59. A crude estimate of the activity of a scene can be gleaned from the motion vectors. In each of the figures the variance (σ) of the horizontal and vertical motion vectors is plotted in the top half of the figures (Figures 58(a) and 59(a)). As expected, the FLOWER GARDEN sequence with simple panning motion has relatively small and constant variation. In turn the FEC allocated to each GOP (Figures 58(b)) is relatively constant. In contrast, the TABLE TENNIS sequence beings with no camera motion followed by a zoom out at Frame 25 (marked by vertical red line) until frame 89 (marked by vertical red line) which is a scene change return to a fixed camera postion. The scene concludes with another scene change at frame 147. The motion vector variance gives an indication of this activity (Figures 59(a)) and as noted by the plot of the FEC allocation (Figures 59(b)), the variable allocation scheme accounts for this increase in activity by allocating additional FEC packets to the appropriate GOP. It can be seen from this example that the variable allocation scheme is able to compensate for variations in the sequence.

7.8 Summary

In this chapter, three improved FEC allocation algorithms that are capable of working in both stationary and non-stationary channel loss conditions have been presented. The results indicate that each of these methods outperforms the standard equal error protection and unequal (fixed) error protection schemes. It was also shown that the variable rate allocation of FEC was able to improve the distortion at the receiver based on variations in

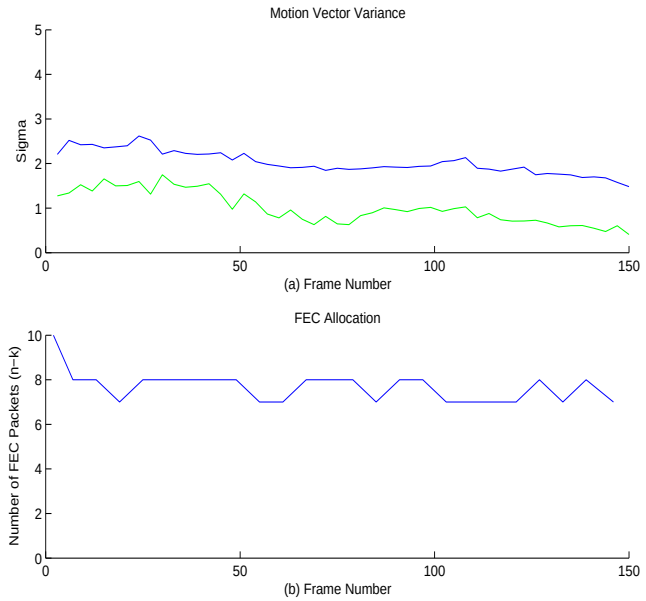


Figure 58: **(a)** Variance of the motion vectors for each frame. **(b)** Allocation of FEC packets for each GOP.

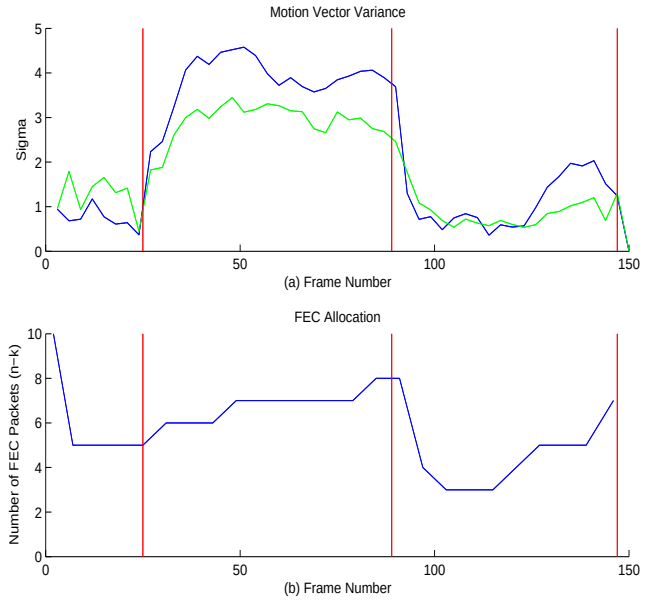


Figure 59: **(a)** Variance of the motion vectors for each frame. **(b)** Allocation of FEC packets for each GOP. Red vertical lines indicate significant changes in camera motion or scene change.

the sequence as well as in response to varying channel QOS. In addition, the results presented here are based on trials with an MPEG-2 codec, but are applicable to any packetized video standard under the constraint that the distortion caused by packet loss can be accurately quantified. This work shows significant improvement in the allocation of FEC. In closing, the results presented here demonstrate that joint source-channel coding techniques can provide improved performance with no additional constraints or demands on the channel bandwith or QOS.

In this chapter, traditional R-S FEC techniques are used to improve the error resilience of the encoded video. In the next chapter, non-traditional forward correction information that assists the decoder in error recovery will be considered.

CHAPTER VIII

FORWARD CONCEALMENT INFORMATION

8.1 Introduction

In the last chapter, Reed-Solomon forward error correction (R-S FEC) was employed to mitigate the effect of packet loss on the bitstream. The advantage of R-S FEC is the complete recovery of all lost data if sufficient FEC is applied; however, if the FEC is insufficient to recover the lost packets its has no value. In addition, it was demonstrated that using enough FEC to eliminate all errors reduced the PSNR at the receiver. Hence, even in optimal allocation of FEC, the decoder must recover from some errors. In previous chapters, the use of certain forward concealment information to aid the decoder in recovering after encountering an error proved useful. In this chapter, the use of this concealment information in conjunction with R-S FEC will be considered. In the first section, the importance of resynchronization information will be considered. In Section 8.3 the addition of concealment vectors will be presented and Section 8.4 will theoretically discuss the use of other types of forward concealment information.

8.2 Resynchronization Information

In the design of streaming bit applications the inclusion of resynchronization codes is a point of compromise. If the decoder loses synchronization then all data received before the next resynch point may be lost; however, resynch points represent overhead that should be minimized for coding efficiency. In most block-based coding algorithms, resynchronization at the macroblock is excessive; therefore, these codecs usually include resynchronization points at the slice or group of blocks (GOB) level. For packet transmission, slices and GOB's do not normally align with packet boundaries, hence the data contained between a lost packet and the next GOB is lost. The importance of recovering this information was presented in Chapters 5 and 6 in which data hiding was used to recover from errors before the

next resynchronization point. The importance of this information is again emphasized by the error resilient (ER) tools of the MPEG-4 codec which allows GOB's to be aligned with packet boundaries and include macroblock (MB) addresses to resynchronize the decoder after a packet loss. In this section, the use of MB level resynchronization as a part of forward error correction will be demonstrated and discussed.

The inclusion of MB level resynchronization data at a packet level can be accomplished with a relatively small amount of overhead; however, it does require coordination between the encoder and the packetization algorithm. Although not complete, resynchronization can usually be accomplished with three additional pieces of information, the location of the next MB, the address of the next MB and either the DCT-DC coefficient for intra-coded MB's or the motion vectors for inter-coded MV's. Additional information such as the quantization level can also be added. In the work presented here, these values can be transmitted using approximately four bytes of data. Using a 256 byte package size to transmit CIF images with a MPEG-2 codec, the location of the next MB can be designated with eight bits (MB's are byte stuffed), the MB address requires nine bits (0–329), which leaves 15 bits to represent the quantization level and the DCT-DC coefficient or the motion vectors. The number of bits to represent the MV is determined by the encoder's motion vector search area, but 10 to 12 bits is normally adequate to represent two vectors, while quantization levels usually require 3 to 5 bits. DCT DC levels quantized to 10 to 12 bits are adequate. The four bytes of data per packet represents approximately 1.5% of the total bandwidth.

To demonstrate the effectiveness of including this type of forward correction information (FCI), the FLOWER GARDEN sequence was encoded with and without the addition of this data. As in the previous chapter, given the constraint $R_S(t) + R_{fec}(t) < R_C(t)$, where $R_C(t)$ is the channel capacity and $R_S(t)$ and $R_{fec}(t)$ are the source rate and the rate of the FEC respectively, the FCI rate reduce $R_S(t)$. The bitstreams were protected by R-S FEC packets using equal error protection (EEP) on each group of packets. The amount of FEC was chosen to provide the smallest expected distortion at the receiver as described in Chapter 7. In Figure 60 the average PSNR of decoded bitstreams is presented as a function

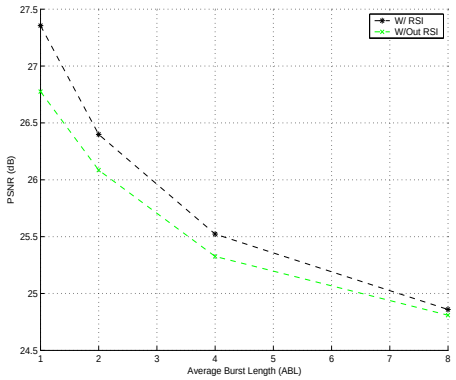


Figure 60: FLOWER GARDEN PSNR vs. ABL with $PLR = 0.10$ using EEP. FEC parameters (n, k) chosen for best expected distortion. Plot demonstrates the value of including resynchronization information in the bitstream.

of average burst length (ABL) for a packet loss rate (PLR) of 10%. Clearly the improvement in PSNR indicates the value of including this resynchronization information. In addition it can be seen that as ABL increases, the value of the additional information decreases. This is the result of fewer errors of longer duration which may extend across several slices; hence the importance of resynchronizing before the end of the slice as a percentage of recovered block decreases. This is shown in Table 19 which presents a macroblock average burst length (versus packet average burst length) which represents the average number of macroblocks lost in each error. The first column is the packet ABL, the second and third column are the MB average burst lengths (MB ABL) with and without resynchronization information and the last column indicates the percent increase in MB ABL without the use of resynch information. As the packet ABL increases, the number of sequential MB's lost increases as expected. The increase in the MB ABL for no resynchronization information ranges from 11.8 to 20.3 with the average increase of 14.4 MB's. The expected value of the increase based on a simple uniform distribution of errors across a 22 MB slice would be 11 MB increase which agrees well with the increase noted here. The percent increase decreases with increase packet ABL, which degrades the value of the resynchronization information. In the single packet loss case, the resynch information decreased the number of lost MB's by 50% while for the $ABL = 8$ experiments the improvement was only 12%. In conclusion, for block-based decoders which do not have resynchronization information

Table 19: Macroblock average burst length for FLOWER GARDEN sequence with $PLR = 0.10$ and FEC parameters (n, k) chosen for best expected distortion. As the average burst length increases, the improvement offered by resynchronization information decreases as demonstrated by the increase in lost macroblocks.

ABL	MB ABL With Resynch Info	MB ABL Without Resynch Info	Change (%)
1	11.65	23.82	51.1%
2	43.02	52.82	18.55%
4	70.03	83.68	16.31%
8	149.62	169.65	11.80%

available (MPEG-1, MPEG-2, H.263) the inclusion of this information represents a minor increase in overhead (or insignificant reduction in error free PSNR for constant bit rate applications) for a significant increase in the received quality across the entire range of ABL's presented here.

8.3 Concealment Vectors

Concealment vectors can enhance the performance of error concealment schemes by providing an optimal vector (based on a specific error criteria) for temporal error concealment. In Chapter 3, several decoder-based error concealment schemes were evaluated to include nearest neighbor statistics such as mean MV's, median MV's, adjacent MV's, and several search techniques. Several error measures based on the surrounding pixels and macroblocks were also evaluated. The final results showed promise in many of these applications; however, the use of the correct MV's (as computed at the encoder using standard MV computations) provided the best error concealment based on a PSNR metric. Based on these findings, this section will evaluate the trade off of concealment motion vectors in a constant bit rate system. Again, assuming a constant $R_C(t)$, $R_S(t)$ and $R_{fec}(t)$ can be varies to provide additional protection for the bitstream. Using this notation, the transmission of concealment motion vectors (CMV) will be considered part of the $R_{fec}(t)$ bit rate which in turn reduces $R_S(t)$. In this section, an estimate of the bandwidth required to transmit the CMV's will be computed and these results will be used to contrast the received image quality with and with out CMV's.

To estimate the bandwidth required to transmit the concealment motion vectors, the entropy of the motion vectors was computed for the three sequences FLOWER GARDEN, TABLE TENNIS, and FOOTBALL. The entropy, H is computed using Equation 60

$$H = -\sum_{i=1}^{L} P_i \log_2 P_i \tag{60}$$

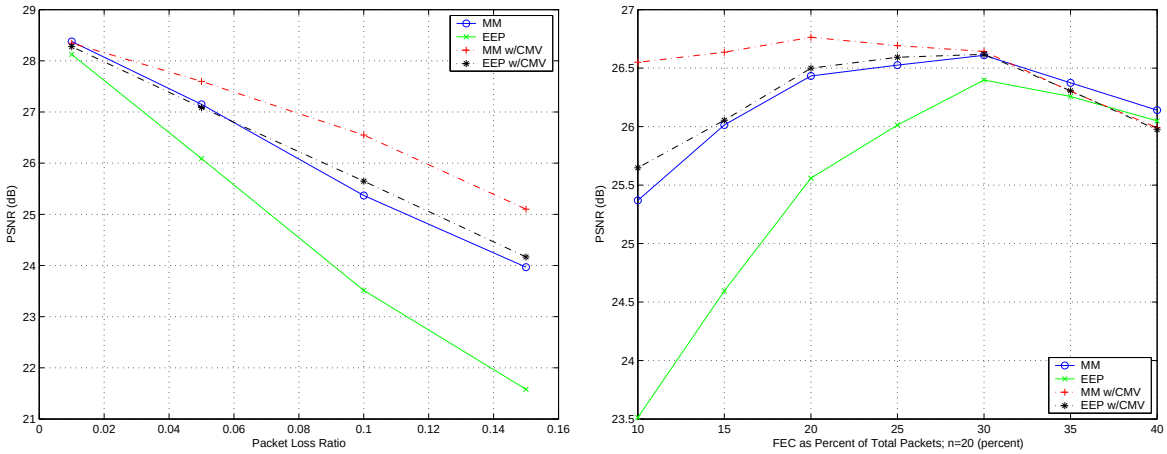
where L is the range of the motion vectors and P_i is the probability that each motion vector has value i. The estimated entropies for the horizontal (x)and vertical (y) motion vectors are based on 50,000 sample motion vectors and are presented Table 20 for the three sequences. The last column of the table is the bit rate in kilobits/sec (Kbps) required to transmit the MV's based on the given entropies.

To compare the effects of error recovery with and without concealment motion vectors, the estimated bit rate for the CMV's (see Table 20) was subtracted from the source rate, $R_S(t)$, in order to maintain a constant channel rate, $R_C(t)$ for a fixed quantity of R-S FEC (i.e. $R_S(t) + R_{fec}(t) + R_{CMV}(t) = R_C(t)$). Figures 61 through 63 contrast the results both with and without CMV's for an equal error protection (EEP) algorithm and the adaptive minimum-maximum (MM) algorithm (Chapter 7). In each set of Figures the left figure plots the PSNR versus PLR with 10% FEC ($n = 20$, $k = 18$) and with the $ABL = 2$. In the right figure, PSNR is plotted versus quantity of FEC applied to each packet. In these cases, the $PLR = 0.10$ and the $ABL = 2$.

In the FLOWER GARDEN sequence, as the PLR increases for a fixed quantity of FEC, the contribution of the adaptive algorithm and the CMV's is evident (Figure 61(a)). Also as the quantity of FEC is varied (Figure 61(b)), the addition of CMV's is able to improve the PSNR of the received sequence in all cases except the 35% and 40% cases. In this regime, the FEC recovers most errors and the contribution of the CMV's is minimized; however,

Table 20: Entropy of motion vectors and estimated bit rate for transmission of the motion vectors for the FLOWER GARDEN, TABLE TENNIS, and FOOTBALL sequences.

Sequence	x Entropy	y Entropy	Bit Rate (Kbps)
FLOWER GARDEN	4.22	1.98	49.10
TABLE TENNIS	3.49	3.12	52.31
FOOTBALL	3.85	3.68	59.64



(a). (b).

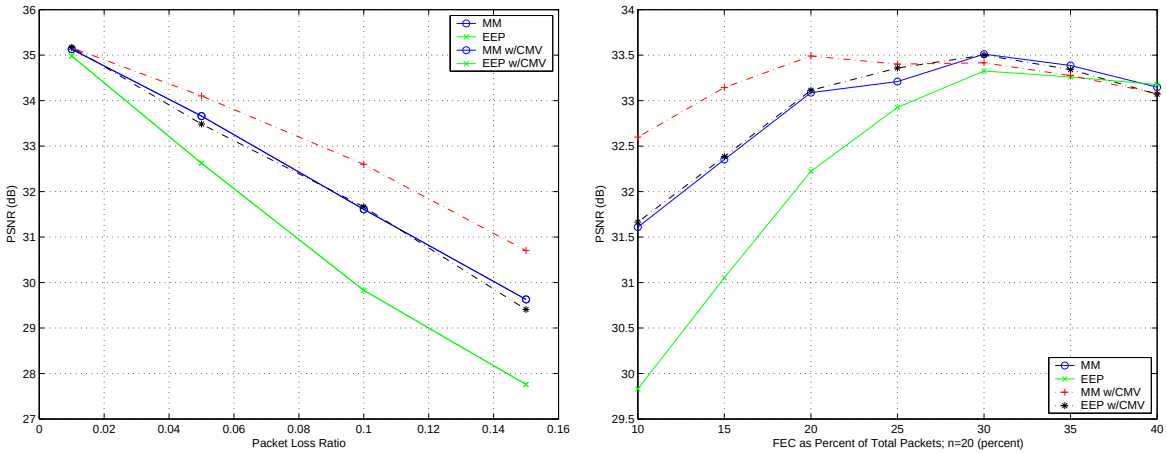
Figure 61: FLOWER GARDEN: **(a)** PSNR vs. PLR with $ABL = 2$ and $n = 20$ and $k = 18$. **(b)** PSNR vs. FEC with $n = 20$, $PLR = 0.10$ and $ABL = 2$.

the bandwidth used to transmit the CMV's is approximately 5% of the source rate which negatively impacts the PSNR of the error-free frames.

Figure 62 presents the results for the TABLE TENNIS sequence, and similar to the FLOWER GARDEN sequence, as the PLR increases for a fixed quantity of FEC, the contribution of the adaptive algorithm and the CMV's is evident (Figure 62(a)). However, as the quantity of FEC is varied (Figure 62(b)), the addition of CMV's is capable of producing similar results to the case without concealment motion vectors, albeit with fewer FEC packets. In these trials the decrease in the source coding to allow for the transmission of the CMV's effectively nullifies the contribution of the CMV's to improved error recovery.

In Figure 63(b) it can be seen that the inclusion of CMV for the FOOTBALL sequence actually has a minor negative effect. A comparison of the optimal rate for FEC (25% for MM w/ CMV and 30% for MM w/out CMV) show a $0.05dB$ difference.

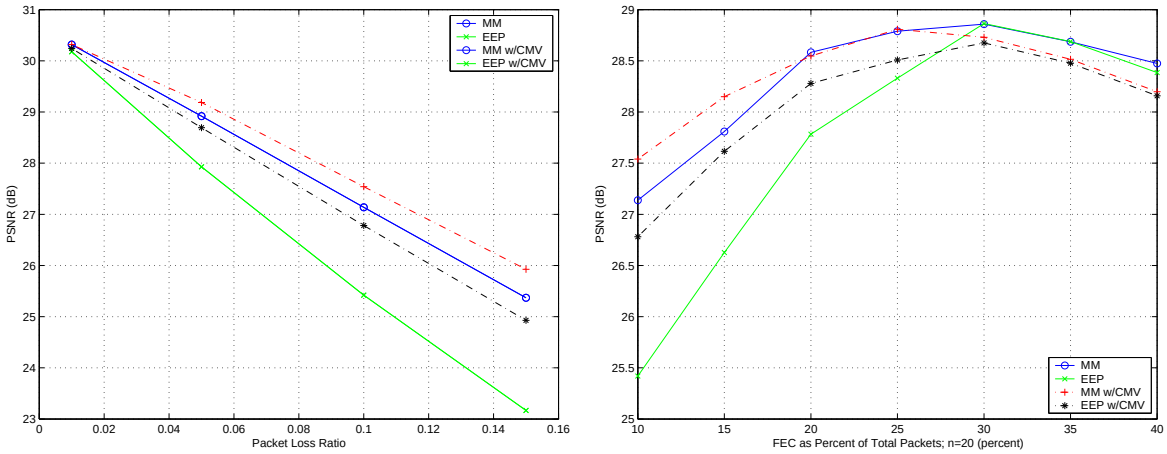
In conclusion, the inclusion of the CMV's cannot always be considered the best operation. In sequences with predictable motion such as the FLOWER GARDEN sequence, the entropy of the CMV's is lower which results in less bandwidth requirements, and the effect of including CMV's is generally positive. As the complexity of motion increases, the inclusion of CMV's may have a minor deleterious effect; however, in all cases, the inclusion of



(a). (b).

Figure 62: TENNIS: **(a)** PSNR vs. PLR with $ABL = 2$ and $n = 20$ and $k = 18$. **(b)** PSNR vs. FEC with $n = 20$, $PLR = 0.10$ and $ABL = 2$.

the CMV's made the decoded distortion less dependent on the FEC rate. For applications in which selecting the ideal FEC rate is difficult, this decreased dependence may be a positive attribute. Additionally, the experiments completed here consider the transmission of CMV's for all frames and the entropy of the CMV's was computed with all CMV's considered as a homogeneous group. A possible method of decreasing the bandwidth required for the CMV's may be to only transmit the CMV's when the frames will be used as a basis for prediction. In our example, this decreases the required bit rate by approximately 66%. Also, the total entropy of the CMV's may be decreased, by considering the CMV's on a per frame basis, which may help align the encoding of the CMV's in relation to the expected motion for different frames. For example, two adjacent frames (I Frame and next B Frame) will have relatively small motion, while frames that are temporally separated (I frame to next P frame) will have a larger range of motion. Using this information to limit the range of CMV's may impact the required bandwidth. In general, the inclusion of CMV's is a trade-off between the bandwidth required to send them and the relative improvement they offer in concealing errors. In the sequences presented here, the effects were usually positive.



(a). (b).

Figure 63: FOOTBALL: **(a)** PSNR vs. PLR with $ABL = 2$ and $n = 20$ and $k = 18$. **(b)** PSNR vs. FEC with $n = 20$, $PLR = 0.10$ and $ABL = 2$.

8.4 Alternative Forward Concealment Information

In the past two sections, two specific types of forward concealment information were considered. The inclusion of resynchronization information in the bitstream requires very little bandwidth and improves error recovery in all cases including long burst length channel conditions. Alternatively, inclusion of concealment motion vectors, requires considerable more bandwidth, but in most cases provided a modest positive improvement. In this section, alternative ideas for forward correction information will be considered.

8.4.1 Encoder-Decision-Based Forward Error Concealment

Motion compensated temporal error concealment is a useful tool for error correction, but has limitations in certain situations. These include scene changes, rapid or deforming motion, zooming, and areas occluded in the previous frame. In these cases, a spatial concealment scheme may provide better results.

Spatial concealment can often yield satisfactory results particularly if directional methods are used; however, it is difficult for the decoder to determine when spatial concealment will outperform temporal concealment. Additionally, as shown in the last section, with

entropy coding, concealment motion vectors require approximately 10 to 12 bits per macroblock. In a system designed by the author, but not fully implemented, the encoder, determines the best type of error concealment, based on a given error measure, and forwards this information to the decoder for error recovery. In this prototype system, the decoder decides among several types of error concealment (See Table 21). The first is none which copies last frame and is equivalent to temporal concealment without motion compensation. The next is bi-directional interpolation which works well in flat areas. The alternative spatial concealment scheme is directional interpolation which would require the transmission of the direction which would be based on the lost macroblock instead of surrounding macroblocks. The final possibility is motion compensated temporal concealment. Table 21 shows a possible implementation using Huffman type coding.

One additional requirement for full implementation of this type of system, is the interleaving of macroblocks. The use of interleaved macroblocks provides an excellent basis for bidirectional and directional concealment schemes even in environments where large blocks of contiguous data may be lost. Additionally, motion compensation for temporal concealment can be based on the four nearest neighbors in an interleaved loss pattern. Furthermore, using this deterministic selection of the vector, the concealment information could offer differential corrections thereby improving the temporal concealment and reducing the bit rate of the concealment motion vectors. The interleaving of macroblocks before transmission significantly improves the performance of both spatial and temporal concealment algorithms.

The system prototype described here may be able to significantly improve error concealment at the decoder and does not require significant computational power. Concealment

Table 21: Notional Huffman coding table for encoder-decision-based forward concealment information.

Huffman Code	Error Concealment Scheme
0	None - Frame copy
10	Bi-directional interpolation
110b...b	Directional interpolation using n bits for direction
1110b...b	Temporal concealment using n bits for motion vectors

motion vectors are already computed for every macroblock except for intra-coded blocks. The computational evaluation of directional and bi-directional concealment algorithms can be minimized with the use of a simple error measure. The most important factor in the implementation of this system is the evaluation of bandwidth requirements for the transmission of the forward correction information.

In this section, additional information to assist the decoder in the recovery of lost macroblocks has been presented. The next section will consider additional information to assist the decoder in the recovery of lost header information.

8.4.2 Header Loss Recovery

Any loss in the video bitstream requires the decoder to regain synchronization. In Section 8.2 it was demonstrated that the addition of resynchronization information on the GOB/slice level was a valuable addition. An area of video transmission that has not been thoroughly researched is the recovery from the loss of other type of headers. Specifically the loss of Sequence, group of picture (GOP) or picture header. This section will consider some alternatives to protection of these valuable portions of the bitstream.

Recovery from the loss of the sequence header code is not practical since the decoding process cannot begin without critical parameters such as horizontal and vertical size. Additionally for video on demand applications, the user can reinitiate the request and for live application, the user will wait for the next sequence header.

The GOP header contains 59 bits of data consisting of the group start code, the time code and two bits for closed GOP and broken link indication. Recovery from the loss of a GOP header can usually be accomplished by the decoder. In the MPEG-2 Test Model 5, the decoder did not track picture type; however, with the addition of approximately two bytes of data (included in sequence start code-user data) the GOP size, N, and I frame to P frame distance, M, can be transferred to the decoder. With these parameters, the decoder is able to track the picture type. Considering the other parameters in the GOP header, the time code is used primarily for video tape recording and the last two bits are used for scene editing within the GOP. Hence, these parameters can be neglected for recovery in most

cases. Therefore, with the addition of N and M to the user data in the sequence header the decoder is able to recover from the loss of the GOP header.

The loss of the picture header can result in a serious loss of synchronization in the decoder. The loss of the picture header can be inferred if the loss of data is limited to less than one frame, because the decoder will find the next slice start code which indicates a slice number less than the last decoded slice number. If the decoder is capable of tracking picture types as indicated above, it can estimate the picture type. To be certain of the picture type would require the retransmission of the picture header. One implementation of this could be a redundant picture header transmitted as side information in the packet containing the following picture header. In this manner, after a loss of synchronization, the decoder can search for the next picture header (while buffering all available bits), read the redundant picture header, decode the buffered bits and conceal the missing macroblocks as necessary. Although a picture header can be 2309 bits in length, most of that comprises alternate quantization matrices. A minimal redundant picture header can be constructed as shown in Table 22. The redundant header size depends on the number of parameters that change from frame to frame. The italicized values indicate values that may be unchanged from previous frames. The redundant header could be constructed using 25 to 43 bits which is approximately 1 Kbit/sec. considering 24 frames/sec.

From this discussion and the data regarding resynchronization information contained in Section 8.2, the inclusion of side information to assist the decoder in recovering from errors is a valuable tool in creating error-resilient codecs. The importance of this information is reiterated by the fact that the latest decoder standards, such as MPEG-4 contain the ability to align packet headers with resynchronization points and the inclusion of the header extension code (HEC) to recover from the loss of a picture header (referred to as video object plane).

Table 22: Redundant picture header information and number of required bits.

Information	Bits
Temporal reference	10
Picture type	3
Variable buffer delay	10
f_codes	8–16
Intra dc precision	2
Picture structure	2
Numerous other flags	10
Total	25–43

8.5 Summary

This chapter offers an alternative to traditional R-S FEC for the protection and recovery streaming video applications over lossy channels. As demonstrated in Chapter 7, the optimal level of R-S FEC that maximizes quality for the viewer will still allow a number of errors to exist. Therefore, recovery and concealment of these errors is still a critical function of the decoder. The work presented here demonstrates that additional information generated by the encoder and transmitted to the decoder as side information or as user information can improve the recovery of the decoder following an error. The inclusion of resynchronization information is always a valuable tool because it requires only a moderate amount of the channel capacity and provides significant improvements especially when the average burst length of the channel is low. The inclusion of concealment motion vectors, requires a considerable amount of the channel and may or may not provide improvements in the recovery process; however, it was not a serious detriment in any of the experiments conducted here. Additional possibilities exist for improved recovery and concealment such as detailed macroblock concealment information and the transmission of redundant headers. Both of these untested ideas offer opportunities for improved error resilience of the bitstream. In conclusion, considering alternatives to traditional FEC can provide for improvements in video quality for the receiver.

CHAPTER IX

CONCLUSION

9.1 Introduction

The thesis which motivated this dissertation was the investigation of forward error correction (FEC) techniques in the encoder and error concealment in the decoder to improve the performance of block-based, motion-compensated, temporal prediction, transform codecs in error-prone environments. In this chapter, the contributions of this work will be summarized and possible areas for further work will be presented. The chapter will conclude with a cost-benifit analysis of techniques presented here.

9.2 Contributions

One of the key aspects of each contribution described here is its compatibility with the MPEG standards, particulary MPEG-1 and MPEG-2. Although they are also compatible with the MPEG-4 standards, some may be redundant due to the error-resilient tools (ERT) included in this standard. The contributions of this dissertation are:

- **Temporal Error Concealment:** A thorough evaluation of temporal error concealment algorithms and error measures relative to different video sequences was conducted. The key contribution with respect to error measures used in motion-compensated, temporal, error-concealment is demonstrating the tradeoff between larger evaluation windows and the importance of spatial locality to the missing macroblock. Additionally, the improvement offered by the correct motion concealment vector, as determined by the encoder, versus decoder search methods can be significant and, in turn, justify their transmission (as concealment vectors) in a rate-constrained, error-prone environment.

- **The Hough Transform and Spatial Error Concealment:** In this spatial error concealment algorithm, the Hough transform is used to detect edges in both foreground and background colors. The improvement offered by the Hough transform versus other directional methods is the ability of the Hough to evaluate numerous possible directions and make a "best" selection. The selection criteria in this work not only evaluates the stongest direction as others do, but also ensures the feaure associated with this directions intersects the missing macroblock. In conjuntion with directional interpolation or directional filtering, this method provides improved edge and texture reproduction.

- **Stegano Codec:** The Stegano codec uses data hiding to transmit three types of error correction information: the length of each macroblock to include byte stuffing values, the final value of differentially-encoded DCT-DC values in I frames and the final value of differentially-encoded motion vectors in P and B frames. The key contributions in this prototype are the use of stegonography to carry error correction information and the importance of critical values in the bitstream. The use of this information allows the decoder to resynchronize more quickly with fewer errors than traditional resynchronization techniques and also allows for perfect recovery of differentially-encoded DCT-DC components and motion vectors. Additionally, the use of steganography to transmit the error correction information allows for the information to be distributed across the entire frame which helps ensure its reception in error-prone environments.

- **Parity Codec:** The Parity codec builds upon the Stegano codec by improving the performance of the codec in the error-free environment while maintaining excellent error recovery capabilities. Using parity information hidden in the bitstream, the decoder resynchronizes with fewer errors than traditional Early Resynchronization decoders. In addition, the decrease in data hiding requirements with respect to the Stegano codec allow the error-free image quality to be maintained.

- **Forward Error Correction (FEC) Allocation:** Several methods to allocate Reed-Solomon (R-S) packet-based forward error correction based on expected distortion

show a significant decrease in end-to-end distrortion (using a PSNR metric) in comparison to equal error protection (EEP) and unequal error protection (UEP) techniques. The encoder is also capable of analyzing channel conditions and varying the source and FEC bit rates to provide improved performance compared to fixed allocation encoders. Although many other authors have demonstrated allocation schemes for FEC in a joint source-channel context, the significant difference in this work was the determination by the encoder of the actual distortion that is generated at the decoder (based on the error concealment scheme implemented in the decoder) if a packet (and, in turn, macroblock and pixel) is lost. This information is then used by the encoder to adjust the source rate of the encoder and the FEC allocation. This is in contrast to others who normally use statistical models to estimate the end-to-end distortion and are limited in the application of error concealment to a single modality (temporal error concealment is most often used).

- **Forward Concealment Information (FCI):** Under the constraints of a constant bit rate, the tradeoff between traditional R-S FEC and alternate forward concealment information is evaluated. The inclusion of resynchronization on a per packet basis shows significant improvements with little additional overhead for all cases considered. The inclusion of concealment motion vectors represents a more significant overhead, but also provides improved quality for the receiver, depending upon the complexity of the video being transmitted. Results for varying types of video are presented.

Although the work presented here is a significant contribution to the state of the art, there are always areas of further research.

9.3 Future Work

Every research project is not an end in itself, but a launching point to investigate other possiblilites. In this section, additional opportunities for research are discussed.

- **Temporal Error Concealment:** The area of temporal error concealment has been investigated thoroughly at this time and its limitations are well documented. The

remaining work in this area should be the evaluation methods to construct better concealment vector fields, or using a combimation of concealment vectors with search methods used by the decoder. Both of these areas should seek to decrease the cost (i.e. bandwidth) associated with the concealment vectors. Improved vector fields should be evaluated on a rate-disrotion basis and the combination of decoder search and concealment vectors may decrease the bandwidth required to transmit the concealment vectors by using differential offsets to the decoder search results. Addtionally, providing the decoder with information that indicates when temporal error concealment is not the best concealment method is another possible way to improve overall performance.

- **The Hough Transform and Spatial Error Concealment:** The work presented here was inspired by the incredible human visual system. We as a species can see and reconstruct images through a number of obscurants such as screens and venetian blinds, fog and haze, day and night. This inspired the use of computer or synthetic vision as a method of reconstructing missing macroblocks. The use of computer vision in the compression and concealment of errors is an area that may provide numerous opportunities for further investigation.

- **Stegano and Parity Codec:** The use of steganography to transmit error correction information has not been completely exhausted, but hiding information in the DCT presents a serious challenge. The fact that the DCT provides excellent compression of digital images makes it difficult to use as a data hiding vehicle when evaluated in a PSNR sense. However, as discussed, the quantization matrix does allow the data to be hidden in the middle frequencies which are least perceptable. The degradation in image quality caused by the transmission of hidden data using the DCT should be evaluated with subjective testing or better peceptual evaluation measures.

- **FEC Allocation:** The best FEC allocation methods presented here were add hoc in nature. Conversely, the optimal method was susceptible to local minima trapping in the search for a global optimum. Further investigation of search methods may

provide even better performance. Such methods may include genetic algorithms or swarm searches.

Additionally, the allocation of the FEC on a time-varying basis depends on an accurate rate-distortion estimate for the sequence. In this work, the values were determined off-line and better methods to determine these values will be critical to implementation in real-time systems.

- **Forward Correction Information:** The area of forward correction information opens a large area to which ingenuity and imagination can be applied. When one leaves the contraints of R-S FEC and uses other methods to provide for correction, a myriad of possibilities exist. One such example is already under research by this author in which the encoder will evaluate the best possible error concealment scheme, be it spatial, temporal, or other, and provide this information to the decoder in conjuntion with the information required to implement the concealment stategy (e.g. concealment motion vectors). Another important improvement would be the evaluation of the transmitting concealment information using a rate-distortion criteria.

Although this is not an all inclusive list of areas for further research, it does provide for numerous areas of addition work.

9.4 Conclusion

In any research project, there are a number of dead ends, poor results and failed ideas that only prove to the build character and the researching skills of the individual. There are also numerous successes which are detailed in the literature, published in dissertations, and evaluated by other colleagues. However, there is one final measure of merit. Is this idea worthy of a patent? Are the improvements significant enough that one should implement them (at some real dollar cost) in future codecs? In retrospect, some of the concepts presented in this thesis do, in the authors opinion, merit this consideration given certain constraints. With respects to the encoder, for an error-prone channel, serious consideration should be given to forward concealment information (FCI). The improvements provided by

additional resynchronization points, differentially encoded values and quantization levels far outweights the cost in terms of bandwidth. This conclusion is supported by the error resilient tools (ERT) available in MPEG-4. Also, the allocation of R-S FEC using an expected distortion criteria provides a significant improvement in end-to-end distortion, at no cost in bandwidth. The only cost associated with such an implementation is the additional computation of the distortion measures and a modest increase in memory required to store the distortion frames. These are small prices to pay given current processor capabilities and memory costs. In the decoder, the ability to resynchornize using the FCI information provided by the encoder is a must. Addtionally, the implemtation of improved spatial concealment schemes such as the Hough transform method, must be evaluated on a frame by frame basis. The evaluation would include the relative success of temporal concealment (e.g. scene change vs. no relative motion) and the longevity of the frame (i.e. I frame vs. B frame). The only cost associated with improved spatial error concealment is the computational requirements, which can be minimized as discussed in Chapter 4. In this author's opinion, these are the applications of this research which warrant an expenditure of materials and time to implement.

REFERENCES

[1] Aign, S., "Error concealment, early resynchronization, and iterative decoding for MPEG-2," in *Proceeding of the IEEE International Conference on Communications*, vol. 3, (Montreal, Can), pp. 1654–1658, June 8–12 1997.

[2] Aign, S. and Fazel, K., "Temporal and spatial error concealment techniques for hierarchial MPEG-2 video codec," in *IEEE International Conference on Communications*, vol. 3, pp. 1778–1783, 18–22 June 1995.

[3] Anderson, R. and Petitcolas, F., "On the limits of steganography," *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 474–81, May 1998.

[4] annd Q. Zhang, J. C., Zhi, W., and Chen, C. W., "An fec-based error control scheme for wireless MPEG-4 video transmission," in *2000 IEEE Wireless Communication and Networking Conference*, vol. 3, (Chicago, IL, USA), pp. 1243–1247, 23–28 September 2000.

[5] Aravind, R., Civanlar, M. R., and Reibman, A. R., "Packet loss resilience of MPEG-2 scalable video coding algorithms," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 426–435, Oct 1996.

[6] Barni, M., Bartolini, F., Cappellini, V., and Piva, A., "A DCT-domain system for robust image watermarking," *Signal Processing*, vol. 66, pp. 357–372, May 1998.

[7] Bartolini, F., Manetti, A., Piva, A., Barni, M., Dugelay, J.-L., and K. Rose, K., "A data hiding approach for correcting errors in h.263 video transmitted over a noisy channel," in *2001 IEEE Fourth Workshop on Multimedia Signal Processing*, (Cannes, France), pp. 65–70, IEEE, 3-5 Oct. 2001 2001.

[8] Bender, W., Gruhl, D., Morimoto, N., and Lu, A., "Techniques for data hiding," *IBM Systems Journal*, vol. 35, no. 3&4, pp. 313–336, 1996.

[9] Biersack, E., "Performance evaluation of forward error correction in an ATM environment," *IEEE Journal on Selected Areas in Communications*, vol. 11, pp. 631–640, May 1993.

[10] Cavusoglu, B., Schonfeld, D., and Ansari, R., "Real-time adaptive forward error correction for MPEG-2 video communication over RTP networks," in *Procedings of the 2003 International Conference on Mutimedia and Expo, ICME*, vol. 3, pp. 261–264, 6–9 July 2003.

[11] Chiang, T. and Zhang, Y.-Q., "A new rate control scheme using quadratic rate distortion model," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 7, pp. 246–250, February 1997.

[12] Cote, G., Shirani, S., and Kissentini, F., "Optimal mode selection and synchronization for robust video communication over error prone networks," *IEEE Journal of Selected Areas of Communication*, vol. 18, pp. 952–965, June 2000.

[13] Ding, W. and Liu, B., "Rate control of MPEG video coding and recording by rate-quantization modeling," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 6, pp. 12–20, February 1996.

[14] Duda, R. and Hart, P., *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[15] Elliot, E. O., "A model of the switched telephone network for data communications," *Bell Systems Technical Journal*, vol. 44, pp. 89–109, January 1965.

[16] Fernandez, C. L., Brasso, A., and Hubaux, J., "Error concealment and early resynchronization techniques for MPEG-2 video streams damaged by transmission over ATM networks," in *Digital video compression: algorithms and technologies*, vol. 2668 of *Proceedings of the SPIE*, pp. 372–383, 31 Jan–2 Feb 1996.

[17] Frossard, P. and Verscheure, O., "AMISP: A complete content-based MPEG-2 error-resilient scheme," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 11, pp. 989–998, September 2001.

[18] Frossard, P. and Verscheure, O., "Joint source/FEC rate selection for quality-optimal MPEG-2 video delivery," *IEEE Transaction on Image Processing*, vol. 10, pp. 1815–1825, December 2001.

[19] Hang, H.-M. and Chen, J.-J., "Source model for transform video coder and its application," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 7, pp. 287–311, April 1997.

[20] Hartung, F. and Girod, B., "Watermarking of compressed and uncompressed video," *Signal Processing*, vol. 66, pp. 283–301, May 1998.

[21] Hasan, M., Sharaf, A., and Marvasti, F., "Novel error concealment techniques for images in ATM environments," vol. 5, (Seattle WA), pp. 2833–2836, May 12–15 1998.

[22] Haskell, P. and Messerschmitt, D., "Resynchronization of motion compensated video affected by atm cell loss," in *Proceeding of the 1992 IEEE International Conference on Acoustic, Speech and Signal Processing*, vol. 3, (San Francisco, CA), pp. 545–548, 1992.

[23] Hsu, Y.-F., Chen, Y.-C., Huang, C.-J., and Sun, M.-J., "MPEG-2 spatial scalable coding and transport stream error concealment for satellite TV broadcast using Ka-band," *IEEE Transactions on Broadcasting*, vol. 44, pp. 77–86, March 1998.

[24] Huang, C.-L. and Liang, S., "A model-driven joint source and channel coder for MPEG-2 video transmission," in *Proceeding of the 2002 IEEE International Conference on Acoustic, Speech and Signal Processing*, vol. 3, pp. 2777 – 2780, 2002.

[25] Johnson, N. F. and Jajodia, S., "Exploring steganography: Seeing the unseen," *IEEE Computer*, vol. 31, pp. 26–34, Feb 19998.

[26] Kanal, L. N. and Sastry, A. R. K., "Models for channels with memory and their applications to error control," *Proceedings of the IEEE*, vol. 66, pp. 724–744, July 1978.

[27] Kim, J., Mersereau, R., and Altunbasak, Y., "Error resilient image and video transmission over the internet using unequal error protection," *IEEE Transactions on Image Processing*, vol. 12, pp. 121–131, Feb 2003.

[28] Kwok, W. and Sun, H., "Multi-directional interpolation for spatial error concealment," *IEEE Transactions on Consumer Electronics*, vol. 39, pp. 455–460, August 1993.

[29] Langelaar, G., Setawan, I., and Lagendijk, R., "Watermarking digital image and video data," *IEEE Signal Processing*, vol. 17, pp. 20–46, September 2000.

[30] LeBuhan, C., "Software-embedded data retrieval and error concealment schemes for MPEG-2 video sequences," in *Digital video compression: algorithms and technologies*, vol. 2668 of *Proceedings of the SPIE*, pp. 384–391, 31 Jan–2 Feb 1996.

[31] Lee, S., Youn, J. S., Jang, S. H., and Jang, S. H., "Transmission error detection, resynchronization, and error concealment for MPEG video coder," in *Proceedings of the SPIE*, vol. 2094, pp. 195–204, 1993.

[32] Lee, S. H., Lee, P. J., and Ansari, R., "Cell loss detection and recovery in variable rate video," in *Proceeding of the 3rd International Workshop on Packtized Video*, (Morristown), March 1990.

[33] Leontaris, A. and Cosman, P. C., "Video compression for lossy packet networks with mode switching and a dual-frame buffer," *IEEE Transactions on Image Processing*, vol. 13, pp. 885–897, July 2004.

[34] Lin, L.-J. and Ortega, A., "Bit-rate control using piecewise approximated rate-distortion characteristics," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 8, pp. 446–459, August 1998.

[35] Mayer-Patel, K., Le, L., and Carle, G., "An MPEG performance model and its application to adaptive forward error correction," in *Proceedings of the ACM International Multimedia Conference and Exhibition*, (Pins, France), pp. 1–10, 1–6 Dec 2002.

[36] Meron, R., Magal, P., and Ianconescu, H., "A robust error resilient video compression algorithm," in *Proceedings of 1994 IEEE MILCOM*, vol. 1, (Fort Manmouth, NJ), pp. 247–251, 2–5 Oct. 1994 1994.

[37] Mitchell, O. R. and Tabatabai, A. J., "Channel error recovery for transform image coding," *IEEE Transactions on Communications*, vol. 29, pp. 1754–1762, Dec 1981.

[38] Mohr, A., Riskin, E., and Ladner, R., "Graceful degradation over packet erasure channels through forward error correction," in *Data Compression Conference*, (Snowbird, UT), pp. 92–101, 29–31 Mar 1999.

[39] Naka, N., Adachi, S., Saigusa, M., and Ohya, T., "Improved error resilience in mobile audio visual communications," in *Proceedings of the IEEE Internatinal Conference on Universal Personal Communications* (1995, ed.), vol. 18, (Tokyo, Japan), pp. 702–706, November.

[40] Ngan, K. N. and Stelle, R., "Enhancement of pcm and dpcm images corrupted by transmission errors," *IEEE Transactions on Communications*, vol. 30, pp. 257–265, Jan 1982.

[41] Orchard, M. T., Wang, Y., and Vaishampayan, V., "Redundancy rate-distortion analysis of multiple description coding using pairwise correlating transforms," in *International Conference on Image Processing(ICIP)*, vol. I, (Santa Barbara, CA), pp. 608–611, October 1997 1997.

[42] Park, J. W., Kim, J. W., and Lee, S. U., "Dct coefficients recovery-based error concealment technique and its application to the MPEG-2 bit stream error," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 845–854, December 1997.

[43] Petitcolas, F. A., Anderson, R. J., and Kuhn, M. G., "Information hiding – a survey," *Proceedings of the IEEE*, vol. 87, pp. 1062–1078, July 1999.

[44] Redmill, D. W. and Kingsbury, N. G., "The EREC: An error-resilient technique for coding variable-length blocks of data," *IEEE Transactions On Image Processing*, vol. 5, no. 4, p. 565, 1996.

[45] Richardson, I. E. G. and Riley, M. J., "Improving the error tolerance of MPEG video by varying slice size," *Signal Processing*, vol. 46, pp. 369–372, October 1995.

[46] Robie, D. and Mersereau, R., "The use of Hough transforms in spatial error concealment," in *Intl. Conf. on Acoustic Speech and Signal Processing*, vol. IV, (Istanbul), pp. 2131–2134, IEEE, 5–9 June 2000.

[47] Rose, K. M. and Heiman, A., "Enhancement of one-dimensional variable-length dpcm images corrupted by transmission errors," *IEEE Transactions on Communications*, vol. 37, pp. 373–379, April 1989.

[48] Shannon, C. E., "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423 and 623–656, July and October 1948.

[49] Smith, J. and Comiskey, B., "Modulation and information hiding in images," in *Proceedings of the First Information Hiding Workshop*, (Cambridge, U.K.), May 1996.

[50] Subbalakshmi, K. P. and Chen, Q., "Joint source-channel decoding for MPEG-4 coded video over wireless channels," in *IASTED International Conference on Wireless and Optical Communication*, (Banff, Canada), July 2002.

[51] Suh, J. W. and Ho, Y. S., "Error concealment based on directional interpolation," *IEEE Transactions on Consumer Electronics*, vol. 43, pp. 295–302, Aug. 1997.

[52] Suj, J.-W. and Ho, Y.-S., "Error concealment technique based on optical flow," *Electronic Letters*, vol. 38, pp. 1020–1021, 29 August 2002.

[53] Sun, H. and Kwok, W., "Concealment of damaged block transform coded images using pojections onto convex sets," *IEEE Transactions on Image Processing*, vol. 4, pp. 470–477, April 1995.

[54] Tsai, I. W. and Huang, C. L., "Hybrid cell loss concelment methods for MPEG-II-based packet video," *Signal Processing: Image Communication*, vol. 9, pp. 99–124, 2 January 1997.

[55] Tsekeridou, S. and Pitas, I., "MPEG-2 error concealment based on block matching principles," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 646–658, June 2000.

[56] Vaishampayan, V. A., "Design of multiple descriptor scalar quantizers," vol. 39, pp. 821–834, May 1993.

[57] Wada, W., "Selective recovery of video packet loss using error concealment," *IEEE Journal of Selected Areas of Communications*, vol. 7, pp. 807–814, June 1989.

[58] Wang, Y., Orchard, M. T., and Reibman, A. R., "Multiple description image for noisy channels by pairing transform coefficients," in *Proceedings 1997 IEEE 1st Workshop on Multimedia Signal Processing*, pp. 419–424, 1997.

[59] Wang, Y., Wenger, S., Wen, J., and Katsaggelos, A. K., "Error resilient video coding techniques," *IEEE Signal Processing Magazine*, vol. 17, pp. 61–82, July 2000.

[60] Wang, Y. and Zhu, Q.-F., "Error control and concealment for video communications: A review," *Proceedings of the IEEE*, vol. 86, pp. 974–977, May 1998.

[61] Wen, J. and Villasenor, J., "Reversible variable length codes for robust image and video coding," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 2, (Santa Barbara, CA), pp. 65–68, Oct 1997.

[62] Wenger, S., "Video redundancy coding in H.263+," in *Proc. AVSPN*, (Aberdeen, UK), September 1997 1997.

[63] Wilson, D. and Ghanbari, M., "Optimization of two-layer SNR scalability for MPEG-2 video," in *Proceeding of the 1997 IEEE International Conference on Acoustic, Speech and Signal Processing, pages = 2637–2640, year = 1997, volume = 4, address = Munich, Germany, month = 21–24 April,*.

[64] Wu, D., Hou, Y. T., Li, B., Zhu, W., Zhang, Y., and Chao, H. J., "An end-to-end approach for optimal mode selection in internet video communication: Theory and application," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 977–995, June 2000.

[65] Zhang, J., Arnold, J. F., and Frater, M. R., "Cell-loss concealment technique for MPEG-2 codec video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 659–665, June 2000.

[66] Zhang, R., Regunathan, S. L., and Rose, K., "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 966–976, June 2000.

[67] Zheng, H. and Boyce, J., "An improved udp protocol for video transmission over internet-to-wireless networks," *IEEE Transactions on Multimedia*, vol. 3, pp. 356–365, September 2001.

[68] Zhu, Q. F., "Device and method of signal loss recovery for real-time and/or interactive communications." U.S. Patent 5550847, August 1996.

[69] Zhu, Q. F., Wang, Y., and Shaw, L., "Coding and cell-loss recovery in dct-based packet video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, no. 3, pp. 248–258, 1993.