UNIVERSITY OF CALIFORNIA
Santa Barbara

Watermark-Based Error Concealment Algorithms for Low Bit Rate Video Communications

A dissertation submitted in partial satisfaction of the requirements for the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

by

Chowdary Balineedu Adsumilli

Committee in charge:

Professor Sanjit K. Mitra, Chair
Professor Jerry D. Gibson
Professor Antonio Ortega
Professor Upamanyu Madhow
Professor John M. Foley

September 2005

UMI Number: 3186818

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

The dissertation of Chowdary Balineedu Adsumilli is approved:

Committee Chair

August 2005

To Appaji, a constant beacon of guiding light.
And to my parents.

ACKNOWLEDGEMENTS

I am very grateful to Professor Sanjit Mitra, chairman of my thesis committee, for his guidance and support during the course of this research. Over these years, he has provided and taught me valuable tools in how to carry my research. I am also very grateful to Professor Jerry Gibson and Professor Antonio Ortega for their valuable guidance through the course of this work. I am thankful to Professor John Foley for the many valuable discussions and for guiding me through the experiments carried on for my research. I would also like to thank Professor Upamanyu Madhow for serving in my doctoral dissertation committee and for his valuable suggestions. I am very thankful to Professor Yong Cheol Kim, who helped me by providing rightful directions and pointers during the course of the research. Thanks are also due to Dr. Takao Nishitani, Professor Yu Hen Hu, and Dr. Ingrid Heynderickx for their comments and critical suggestions.

I thank the current and previous members of the Image Processing Research and Computer Vision Laboratories: Myléne Farias, Chin Koh, John Berger, Marty Ho, Gabriel Gomes, Mike Moore, Christian Schmidt, Kaushal Solanki, Ken Sullivan, Sitaram Bhagavati, and Marco Zuliani. They have all made my life and work at UCSB an enjoyable experience. I also thank the visitors we had in the lab from time to time: Fabio Tonci, Elisa Gelasca, Helmut Singer, Kenichi Iwatha, and Moises Rebierio for making my stay in the lab more memorable. I am very thankful to Marco Carli, with whom I have collaborated in many papers. I am also very thankful to Valerie DeVeyra, for the help with the administrative work.

I would like to thank Appaji, my parents, and my sister for their constant support and encouragement throughout the process of my PhD and for providing the right words of motivation when most required.

This work was supported in part by a National Science Foundation Grant CCR-0105404 and in part by a University of California MICRO Grant with matching support from Intel Corporation, Microsoft Corporation, NEC Corporation, and Philips Research Laboratories.

VITA

November 1976	Born, Hyderabad, India.
January 1999 – June 1999	Undergraduate Researcher Navigational Electronics Research & Training Unit (NERTU), Hyderabad, India.
June 1999	**Bachelor of Technology** Dept. of Electronics & Communications Engr. Jawaharlal Nehru Technological Univ., India.
May 2000 – August 2000	Employed as Summer Intern Santa Clara group, Sprint PCS Inc. Pleasanton, California.
January 2000 - May 2001	Graduate Teaching Assistant Dept. of Computer Science & Statistics University of Wisconsin, Madison.
May 2001 – August 2001	Employed as EID Engineering Intern GEMnet Group, GE Medical Systems - IT Menomonee Falls, Wisconsin.
December 2001	**Master of Science** Dept. of Electrical & Computer Engineering University of Wisconsin, Madison.
September 2001 – June 2002	Graduate Researcher Dept. of Electrical & Computer Engineering University of Wisconsin, Madison.
September 2003 – December 2003	Employed as a Research Intern Video Processing and Visual Perception Group Philips Research Laboratories (NATLAB) Eindhoven, The Netherlands.
July 2002 – June 2005	Graduate Research Assistant Dept. of Electrical & Computer Engineering University of California, Santa Barbara.
September 2005	**Doctor of Philosophy** Dept. of Electrical & Computer Engineering University of California, Santa Barbara.

PUBLICATIONS

[1] Chowdary B. Adsumilli, Cabir Vural, and Damon L. Tull, “A noise based quantization model for restoring block transform compressed images,” *Proc. IASTED Intl. Conf. on Signal and Image Processing*, Hawaii, USA, August 2001, pp. 354-359.

[2] Chowdary B. Adsumilli and Yu H. Hu, “Adaptive wireless video communications: Challenges and approaches,” *Proc. Intl. Workshop on Packet Video*, Pittsburgh, Pennsylvania, USA, April 2002.

[3] Chowdary B. Adsumilli and Yu H. Hu, “A dynamically adaptive constrained unequal error protection scheme for video transmission over wireless channels,” *Proc. IEEE Intl. Workshop on Multimedia Signal Processing*, Virgin Islands, USA, December 2002, pp. 41-44.

[4] Chowdary B. Adsumilli, Mylene C.Q. Farias, Marco Carli, and Sanjit K. Mitra, “A hybrid constrained unequal error protection and data hiding scheme for packet video transmission,” *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, Hong Kong, April 2003, pp. 680-683.

[5] Chowdary B. Adsumilli, Mylene C.Q. Farias, Marco Carli, and Sanjit K. Mitra, “A robust error concealment technique using data hiding for image and video transmission over lossy channels,” *IEEE Trans. on Circuits and Systems for Video Technology*, 2005, In press.

[6] Chowdary B. Adsumilli and Sanjit K. Mitra, “Error concealment in video communications using DPCM bit stream embedding,” *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, Philadelphia, USA, April 2005, pp. 169-172.

[7] Chowdary B. Adsumilli, Sanjit K. Mitra, and Yong C. Kim, “Detector performance analysis of watermark-based error concealment in image communications,” *Proc. IEEE Intl. Conf. on Image Processing*, In press.

[8] Marco Carli, Chowdary B. Adsumilli, Valerio Razautti, and Alessan-dro Neri, “Video watermarking in 3D DCT domain,” *Proc. Intl. Workshop on Spectral Methods and Multirate Signal Processing*, Riga, Latvia, June 2005, Invited paper.

[9] Fabio Tonci, Chowdary B. Adsumilli, Marco Carli, Alessandro Neri, and Sanjit K. Mitra, “Buffer constraints for rate-distortion optimization in mobile video communications,” *Proc. Intl. Symp. on Signals, Circuits and Systems*, Lasi, Romania, July 2005, Invited paper.

ABSTRACT

Watermark-Based Error Concealment Algorithms for Low Bit Rate Video Communications

by

Chowdary B. Adsumilli

In this work, a novel set of robust watermark-based error concealment (WEC) algorithms are proposed. Watermarking is used to introduce redundancy to the transmitted data with little or no increase in its bit rate during transmission. The proposed algorithms involve generating a low resolution version of a video frame and seamlessly embedding it as a watermark in the frame itself during encoding. At the receiver, the watermark is extracted from the reconstructed frame and the lost information is recovered using the extracted watermark signal, thus enhancing its perceptual quality. Three DCT-based spread spectrum watermark embedding techniques are presented in this work. The first technique uses a multiplicative Gaussian pseudo-noise with a pre-defined spreading gain and fixed chip rate. The second one is its adaptively scaled version and the third technique uses informed watermarking. Two versions of the low resolution reference, a halftoned reference and a DPCM encoded reference, are considered here.

Spatial, temporal, and spatio-temporal implementations of WEC are proposed for video. A reference watermark of either the intra-coded frame or the subsequent inter-coded frame is embedded in the current frame for mitigation of error propagation. In the case of spatio-temporal implementation, a low resolution gray-scale reference is bit-plane embedded in a volume data set. These implementation schemes not only enhance the end-user perceptual video quality, but also increase the embedding capacity.

Both qualitative and quantitative analysis of the WEC algorithms along with a comparison between the full-frame and block-based embedding techniques are presented. A psychophysical experiment is performed to obtain the subjective quality evaluation of the proposed techniques, the comparison between the perceptual quality of intra-coded reference embedding and inter-coded reference embedding, and to verify the codec-independency of the WEC algorithms.

Experimental results show that the proposed WEC techniques outperform other current error concealment algorithms, especially at higher transmission losses. In video implementations, the inter-coded frame embedding resulted in nearly a constant perceptual quality performance when compared to the intra-coded frame embedding scheme. The psychophysical experimental analysis has confirmed this observation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Error control in wireless video communications is of primary importance in successful transmission and reception of image/video signals over bandwidth limited wireless networks or fading communication channels. Pre- and post-processing error control mechanisms like error resilience and error concealment have been developed and incorporated in the design of the basic communication structure to make the data more robust to wireless fading channel errors in existing video transmission standards like H.263(++), H.264, and MPEG-x [1]-[9]. Of high concern is the bridge between the application level QoS requirements of real-time video transmission and the QoS guarantees provided by the wireless channel/networks [10],[11].

Most of the errors occurring in the transmission of real-time video are due to its large bandwidth requirements and/or restricted allocation of the channel bandwidth. Error resilience at the encoder and error concealment techniques at the decoder have been proposed to minimize the effects of these errors [12]-[19]. However, to meet the QoS guarantees, we believe that their interaction is key in the design of a high perceptual quality wireless video codec [20].

The process of source encoding involves efficiently converting the input original image/video signal into a sequence of bits and compressing them to near entropy levels [13]. The purpose of channel encoder is to introduce redundancy in the compressed bit-stream. It is this process that effectively provides error resilience. At the receiver, the errors are detected as a part of the channel decoding process [21]. It is near the source decoding and reconstruction part where the error concealment techniques help in reducing or hiding these errors [22],[23].

Even though a number of effective error control algorithms have been proposed, they are associated with multiple problems with regard to their implementations. First, they are fragmented and not integrated. In each step of the video communication system, a local optimum is sought rather than a global one in that the efforts to minimize channel errors are independent at the transmitter and receiver ends. Second, these efforts are open ended giving way to overhead when each of the techniques considers its best optimum performance. In such a scenario, the performance of the integrated system is jeopardized to match the independent individual units' optima. And last, these techniques are non-adaptive with regard to bandwidth, display and viewing conditions, and application types.

These problems have motivated the development of an integrated adaptive approach which requires an end-to-end QoS criteria based on visual quality. This approach should aim at having maximum possible perceptual quality at the viewer's end with any possible channel conditions by jointly considering error resilience, channel characterization, concealment/post processing techniques. The implementation of such a codec strategy, which is considered here, would not only involve effectively monitoring the channel conditions and adapting to its variations but also keeping the visual quality within a preferred margin of error.

Subjective tests have proven that even though an increase in channel bandwidth after the start of transmission improves the visual quality of the received video, it also produces a visible variation in the quality which leads to perceivable annoyance. More often than not, the viewer rates a constant quality video (even though this constant is not maximum achievable quality) to be the better than a varying quality one (even though it reaches maximum quality multiple times through out the video). It is for this reason that it is desirable to have nearly constant quality [24].

The basic idea behind this research is towards the design of a wireless video codec that aims at achieving Constant Perceptual Quality (CPQ) using interactive error control at the end user. It involves developing algorithms at both ends for error resilience and concealment, and optimizing them subject to the network bandwidth, latency, and received video quality constraints. The techniques proposed here are based on the underlying assumption that a typical non-feedback based interactive error control is followed both at the receiver and the transmitter [25],[26].

1.1 Problems Addressed

Of the multiple problems described above, this work address a specific set of key issues that hinder the performance of the error control in the integrated end-to-end video communications system. This set contains of three problems each of which represents a different aspect of the error resilience and concealment techniques proposed so far.

A major concern with the existing error resilience and concealment techniques is that the proposed algorithms make the currently existing video codecs non-backwards compatible. Any new or existing error resilience tool or concealment tool would require the codec to be modified and therefore would make it not work with any existing previous versions of the codec [27].

Another problem with the current techniques is that to keep the bit rate constant, they require higher compression (and source coding) from the encoder to incorporate the associated bits necessary for error resilience or concealment. This would not only incur higher complexity (due to higher amount of compression) but also end up producing more compression defects [28], [29].

Alternatively, the resilient or concealment bits could be sent as side information rather than reducing the bit rate required for compressing and encoding the source video. This however, increases the bit rate of the transmitted signal thereby incurring more possibility of packet losses over the channel (caused due to congestion as a result of increased bandwidth)[1]. Moreover, the increased bit rate due to side information is susceptible to packet losses as well thereby reducing the robustness of the resilience or concealment bits [30].

Watermark-based error concealment (WEC) algorithms overcome these problems as embedding a reference produces little or no increase in the bit rate, they are codec-independent and therefore could be made backwards compatible, they spread over a set of frequencies and therefore are robust to channel errors, and they do not require higher

[1] The advantages and disadvantages of using the side information scheme are discussed in more detail in Chapter 3

amount of compression as the error resilience and concealment bits that it transmits are embedded inside the video itself. For these reasons, WEC algorithms work towards achieving a higher error control performance when compared to the existing error control techniques.

1.2 Objective

The main goals and objectives of the WEC algorithms are as follows:

1.2.1 Transmission Error Concealment

The purpose of developing the WEC algorithms has been to remove, reduce, or hide the errors that occur during the transmission of video over lossy wired and wireless channels or to mitigate any error propagation that occur as a result. An additional objective for WEC algorithms is that, concealment of the transmission errors should be achieved without increasing the bit rate or complexity or decreasing the quality of the source video either by higher compression or by additionally introduced artifacts during the process of watermarking. However, to this end, the WEC algorithms eliminate one of the key concerns of backwards incompatibility of the existing codecs when new codec-dependent error resilience and concealment algorithms are incorporated in them.

1.2.2 Optimal Complexity-Bit Rate-Quality Performance

One of the issues that WEC addresses is that the existing error control techniques require either higher compression to maintain a constant bit rate or higher bit rate to maintain a constant level of compression. Consider for example, passing the error concealment information through side information. When side information, typically consisting of redundancy bits for error resilience and motion vectors, is added to the existing compressed source video bits, the bit rate increases. In order to keep the bit rate constant, it requires higher compression of the source. This increase in the amount of compression not only increases the complexity of the transceiver but also increases the strength of the artifacts generated at higher compression thereby decreasing the quality of the source video.

We therefore require algorithms that achieve a more optimal point in the complexity-bit rate-quality performance. The objective of the WEC algorithms is precisely to achieve this point. They eliminate this problem with existing error control techniques by embedding the side information as a watermark with little or no change to the bit rate, thus not requiring higher compression of the source. Also, the artifacts generated due to the watermark embedding process are very few and seldom visible. Therefore, the perceptual quality of the watermark image is quite close to that of the image without the watermark. These advantages of WEC algorithms lead to a better, more optimal performance point in the complexity-bit rate-quality triangle.

1.2.3 Psychophysical Evaluation

Evaluation of the perceptual performance of the WEC algorithms is important to assess the advantages provided by them. We first compare the objective measure (PSNR) values of the WEC algorithms with those of other techniques. However, as subjective tests previously suggested, PSNR is not the right measure to assess the performance of any algorithm. Also, presented in Chapter 4 are two implementations to video that give similar PSNR values to error concealed video, but are perceptually different. With these aspects in consideration, we perform a psychophysical experiment to test not only the performance of the WEC algorithms for varying losses, but also to verify their codec independency. The subjective experiment has the following objectives:

- The subjective evaluation of the increment in the perceptual quality that the proposed technique provides over conventional error concealment in low bit rate video codecs,

- The comparison between the perceptual quality of intra-coded reference embedding in the intra-coded frame and inter-coded reference embedding in the intra-coded frame, and

- The verification of codec-independency during implementation of the proposed algorithm (two low bit rate codecs, MPEG-4 and H.264 have been used for this test).

1.3 Approach

In this dissertation, a set of robust error concealment algorithms are proposed and developed that make use of watermarking to hide a low resolution version of the original image/video frame in itself. This watermark then acts like a reference at the receiver to conceal errors and reconstruct a better quality image/video frame.

The data hiding techniques add redundancy to the transmitted video sequence frame data without increasing its bit rate during transmission. The basic approach consists of independently embedding the original information from the video frames into the data stream as hidden or embedded data. At the receiver, this embedded data is extracted and it provides additional information about the received frame and can be used for detecting and concealing errors. It is important to underline that the proposed techniques do not overload the communication channel by requiring feedback communication or any retransmission of damaged blocks.

It should be noted here that the WEC algorithms can be used for both wired and wireless cases since error resilience can be implemented adaptively with the proposed techniques such that higher channel protection (similar to unequal error protection (UEP) scheme) can be given to the watermarked bits of the encoded image/video frame. This way the reference is extracted with little or no loss at the receiver. This not only helps in the reconstruction of the original frame but also will give us a good estimate of the channel errors for the packets transmitted with UEP.

Dithering techniques have been used to obtain a binary watermark from the low resolution version of the image/video frame along with DPCM bit stream encoding techniques. In the case when dithering techniques are used, multiple copies of the dithered watermark are embedded in frequencies in a specific range to make it more robust to channel errors. It is shown experimentally that based on the frequency selection and scaling factor variation, the watermark can be extracted with high quality even from a low quality lossy received image/video frame. Furthermore, these techniques are compared to an alternative approach where the low resolution version is encoded and transmitted as side information instead of embedding it. Simulation results show that WEC algorithms outperform existing approaches in improving the perceptual quality, especially in the case of higher loss probabilities.

The proposed WEC algorithms have been implemented in the key frames in different spatial and temporal ways. The generated watermark is varied to represent either the current frame or the third subsequent frame thereby creating mechanism to mitigate any error propagation that the key-frame operation involves. A psychophysical experiment has been later carried out in the Psychology Department at UCSB to evaluate and assess the quality improvements in the error concealed video due to WEC algorithms, and to verify the codec independency of the these proposed algorithms. In doing so, we have found the variation of the WEC algorithms' performance due to varying packet loss percentages and have observed one WEC algorithm's performance to have resulted in a constant perceptual quality of the video at the end user.

1.4 Summary of Contributions

The main contributions of this dissertation are:

- Development of a set of a novel watermark-based error concealment techniques. The generated watermark is a low resolution version of the frame itself and helps recover the information losses that occur during the transmission of the video. The binary watermark could be a halftoned version or an encoded bit stream of the low resolution version of the frame. The implementation in case of color components and varied channel losses show the enhanced performance of the developed algorithms.

- Development of the low bit error rate informed watermarking scheme that embeds a copy of the watermark detector inside the encoder. The scale factors as per the coefficients have been increased by minimizing the detector's BER performance as well as maximizing the embedded frame quality.

- Development of different spatial video implementations with two of the key techniques being embedding the I-frame reference in itself and embedding the subsequent P-frame in the current I-frame. The idea of the latter technique emanated from the concepts of constant subjective quality preferences.

- Development of a combined spatio-temporal watermark embedding technique based on 3D-DCT. A gray level reference of the watermark is embedded in the volume cuboid element and it has been shown that higher levels of error concealment could be achieved using this approach.

- Development of a information theoretic approach to the watermark-based error concealment algorithms. The approach initially required an analysis of the performance of the watermark detector and calculation of the probability of error. Based on the overall estimate of the distortion due to this probability of error, the R-D performance of the WEC algorithms has been analyzed and compared to substantial packet loss scenarios.

- Evaluation of the subjective quality increment due to WEC over conventional low bit rate codecs such as MPEG-4 and H.264 by conducting a psychophysical experiment. The experiments have also verified the codec independent operation of the WEC along with evaluating the variation in quality of the compressed videos due to varying channel loss rates. The two spatial implementations of the WEC discussed in chapter 4 along with the baseline error concealment in the codecs are used in the experiment for comparative quality assessment.

1.5 Dissertation Outline

The dissertation is organized in the following manner:

Chapter 2 provides a brief background on the error control that exists in conventional algorithms. The error control techniques at the encoder, called the error resilience techniques, which make the transmitted signal more robust to the channel errors, are explained along with the error concealment algorithms, which reduce or hide the errors that occur during the transmission of the video signal. The need for an integrated end-to-end error control has been substantiated and a possible adaptive solution to the problem of constrained optimization of UEP has been proposed. Sample results illustrate the improvement in the end user perceptual quality of the images due to the proposed optimization.

Chapter 3 describes the different WEC techniques proposed in this work. The basis of the WEC is laid out while explaining the first of these WEC techniques which involves low-resolution watermarking and halftoning. Another technique based on encoded watermark embedding is then proposed. Various cases have been studied and extensions explained in detail along with extensive analysis of the obtained results.

The process of watermarking in both of these scenarios is based on a spread-spectrum algorithm. Based on the detection performance of the watermark detector, a new algorithm, called the informed WEC technique, is developed by incorporating a copy of the detector inside the encoder. This technique not only reduces the BER of the watermark detection process, but also improves the PSNR performance of the extracted reference.

Chapter 4 describes the video implementations of the WEC. Typically since more information can be hidden in video (than an image) without the viewer noticing it, we can adapt the WEC to video in an intelligent fashion by making use of the spatial and temporal redundancy and coding mechanisms. Three broadly classified techniques have been proposed that rely on spatial, temporal, and a combined spatio-temporal embedding. The results show a substantial improvement in performance compared to conventional video error concealment methods.

Chapter 5 outlines the information theoretic approach to the WEC algorithms in brief. The approach is based on the encoder and the detector performance and how the BER is reduced with the WEC. Furthermore, the rate-distortion performance of the WEC is analyzed and it is shown in this chapter that even though the entropy levels increased, the WEC algorithms give a more optimal R-D curve when compared to other error concealment algorithms.

Two spatial video implementations are considered in Chapter 6 for a psychophysical experiment conducted to measure the subjective enhancement in performance due to the WEC. The implementations include embedding I-frame in itself and embedding a reference for the subsequent P-frame in the current I-frame, and are incorporated in both H.264 and MPEG-4 to verify codec independent operation of the WEC. The experiment has been performed using a dual-stimulus comparison as per the ITU-T standards [31]. The conclusions depict an subjective quality improvement when WEC (specifically P-frame reference in I-frame) is implemented on top of conventional codecs.

Chapter 7 draws conclusions on the proposed WEC algorithms, its implementations and the obtained results. It then analyzes the varied types of applications that these algorithms could be used in, and concludes by giving a set of plausible future directions that they make take. We also propose at the end of the chapter, a couple of new ideas that the proposed WEC could be extended to.

The dissertation also includes two appendices that throw light on different concepts introduced in Chapters 4 and 6. Appendix A explains the characteristics of the LCD display used including the calculation of the display γ, while Appendix B lists the instructions given to the subject during the experiment.

Chapter 2

Error Control in Video Communications

The advent of second and higher generation wireless personal communication standards in recent years has made it possible to realize bandwidth efficient video communications. The variation of wireless channel/ network capacity with mobility suggests that substantial performance gains are promised by intelligent multi-mode adaptive transceivers [32], [33], [34]. Specially designed error-resilient, fixed but programmable rate video codecs, which generate a constant number of bits per video irrespective of the video motion activity, provide wireless video communication services [35] over low rate, low latency interactive schemes, and multimedia/videophony applications.

Fig. 2.1 shows the basic structure of a digital wireless video communication system requiring five fundamental operations [13]. The process of source encoding involves efficiently converting the input original image/video signal into a sequence of bits and compressing them to near entropy levels. The purpose of channel encoder is to introduce redundancy in the compressed bit-stream which in turn is used at the receiver to overcome the effects of signal transmission through noisy wireless networks/channels.

The encoded data bit-stream, now segmented into fixed or variable length packets, is modulated and sent over the channel/network. The channel can be assumed as a physical medium, free space/atmosphere in case of wireless, that distorts, fades, and corrupts any information transmission. At the receiving end of a digital communication system, the channel corrupted received signal is demodulated into a data bit-stream, channel decoded

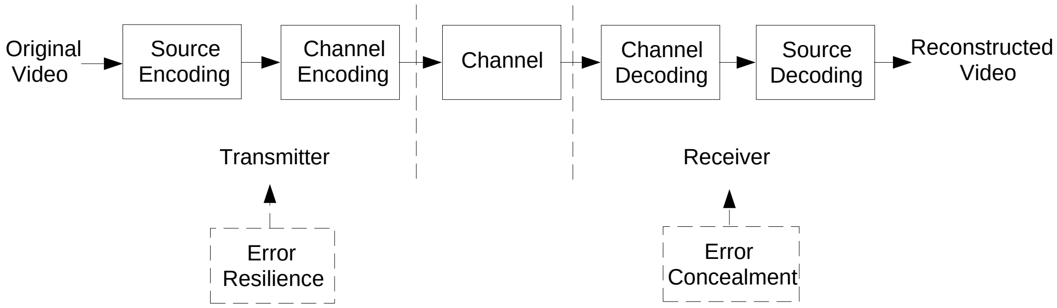


Figure 2.1: Basic Structure of a digital video communication system.

(the decoder attempts to reconstruct the original information sequence from knowledge of the code used by channel encoder and the redundancy contained in the received data) and source decoded to obtain the reconstructed video.

Most of the errors occurring in the transmission of real-time video are due to its large bandwidth requirements. Even if the channel capacity exceeds the required bit rate (high bandwidth channel/low bandwidth application), channel errors can severely degrade the signal and so, a compression scheme [21] and bit rate must be carefully chosen to match the channel characteristics while maximizing the video quality. Error resilience [13] at the encoder, error concealment [17] techniques at the decoder, shown in Fig. 2.1 are discussed further in detail.

In this chapter, we provide an overview of the error resilience and the error concealment involved in the successful transmission of image and video over lossy wired and wireless channels.[1]

2.1 Error Resilience

Error resilience schemes [3], [13], [21] address the issue of compression loss recovery and specifically, they attempt to prevent error propagation by limiting the scope of the damage caused by bit errors and packet losses on the compression layer. The standard error resilient tools include re-synchronization marking, data partitioning, and data recovery. Based on the role that the encoder, the decoder or the network layer plays in the process,

[1] Since wireless channels introduce artifacts that are more complex and intricate to remove, we rate the wireless channel errors to supersede (and in some ways form a superset of) the errors that occur during the transmission through wired channels

these error resilience techniques can be divided into four categories, each of which is described here. A key assumption here is that the video is coded using the block-based hybrid encoding framework.

2.1.1 Robust Entropy Encoding

The encoder in this approach operates to minimize the adverse effects of the transmission errors of the coded bit-stream on the decoder operation so that unacceptable distortions in the reconstructed video quality can be avoided. Compared to coders that are optimized for coding efficiency, error resilient coders are typically less efficient in that they use more bits to obtain the same video quality in the absence of any transmission errors. These extra bits, or redundancy bits, are introduced to enhance the video quality when the bit stream is corrupted by transmission errors.

The sensitivity of a compressed video stream to transmission errors is mainly due to the fact that a video coder uses variable length coding (VLC) to represent various symbols. Any bit errors or lost bits in the middle of the code word not only makes this code word un-decodable but also makes the following code words un-decodable, even if these bits were received correctly. The design goal in the error resilient coders is to achieve a maximum gain in error resilience with the smallest amount of redundancy. Techniques to introduce such redundancy in the bit-stream include:

- *Re-synchronization markers*: Inserting re-synchronization markers periodically enhances the efficiency of encoder error resilience. These markers are designed to be effectively distinguished from other code words and small perturbations of these code words. Header information regarding the spatial and temporal locations or other in-picture predictive information concerning the subsequent bits is attached immediately after the re-synchronization information. The decoder can then resume proper decoding upon the detection of the re-synchronization marker.

  Synchronous markers' utility interrupts in-picture prediction mechanisms like motion vector (MV) or DC coefficient prediction, which in turn adds more bits. Longer and more frequently inserted markers would enable the decoder to regain faster synchronization such that the transmission errors affect a smaller region in the reconstructed frame and so long re-synchronization code words are used in the current video coding systems.

- *Reverse variable length coding* (*RVLC*): With RVLC, the decoder can not only decode bits after a synchronization code word, but also decode the bits before the next synchronization code word, from the backward direction and so fewer correctly received bits will be discarded and the effected area by a transmission error will be reduced. RVLC is adopted in both MPEG-4 and H.263 in conjunction with insertion of synchronization markers. For video coding and applications, RVLC can be designed with near perfect entropy coding efficiency in addition to providing error resilience.

- *Error-resilient entropy code* (*EREC*): EREC is an alternative way of providing synchronization which works by re-arranging variable length blocks into fixed length slots of data prior to transmission. The EREC is applicable to coding schemes where the input signal is split into blocks and these blocks are coded using variable-length codes, each of which is a prefixed code, like the macro-blocks (MBs) in H.263.

2.1.2 Unequal Error Protection (UEP)/Layered Coding

Layered coding (LC) or scalable coding refers to coding a video into a base layer, which provides a low but acceptable level of quality, and one or several enhancement layers, each of which will incrementally improve the quality. LC is a way to enable users with different bandwidth capacity of decoding powers to access the same video at different quality levels. To serve as an error resilient tool, LC must be paired with UEP in the transport system, so that the base layer is protected more strongly.

There are many ways to divide a video signal into two or more layers in the standard block-based hybrid video coder. A video can be temporally down-sampled, and the base layer can include the bit stream for the low frame-rate video, whereas the enhancement layer(s) can include the error between the original video and that up-sampled from the low frame-rate coded video. The same approach can be applied to the spatial resolution, so that the base layer contains a small frame-size video. The base layer can also encode the DCT coefficients of each block with a coarse quantizer, leaving the fine details (the error between the original and the coarsely quantized value) to be specified in the enhancement layer(s). The base layer may then include the header and motion information, leaving the remaining information for the enhancement layer. In the MPEG and H.263 techniques, the first three options are temporal, spatial, and SNR scalability, respectively, and the last one is data partitioning.

In all the approaches discussed here, including UEP and LC, a local minima is sought and these techniques would operate more effectively when the proposed integrated approach is employed.

2.2 Channel Issues

With highly scalable video compression schemes [36], it is possible to generate one compressed bit-stream such that different subsets of the stream correspond to the compressed version of the same video sequence at different rates. Such a source coding algorithm would not have to be altered with varying wireless network/channel conditions. This is particularly attractive in heterogeneous multicast networks where the wireless link is only a part of a larger network and the source rate cannot be adapted to the individual receiver at the wireless node.

2.2.1 Channel Coding

Although channel encoding stage typically uses forward error correction (FEC) codes, the highest coding gain (near entropy level) for Rayleigh fading additive white Gaussian noise (AWGN) channels is achieved using Trellis coded modulation (TCM). Due to variation in importance of different bits within a bit-stream, protection of source bits using UEP schemes is critical to further enhance the performance of the channel encoder. TCM schemes proposed for fading mobile channels provide unequal source sensitivity matched error protection when compared to sequential FEC coding and modulation.

Conventional block and convolution codes are successfully used to combat the bursty errors of fading noisy wireless channels/networks. Rate Compatible Punctured Convolution codecs (RCPCs), which implements UEP, provide bit sensitivity matched FEC protection for sub-band codecs where some of the encoded output bits can be removed or "punctured" from the bit-stream. A variety of different rate bit protection schemes can be designed using the same decoder and protecting the more error sensitive bits by a stronger low rate code and the more robust source coded bits by a higher rate, less powerful FEC code.

Although both source and channel encoding work towards the goal of making the data more resilient to channel errors, source encoding attempts to compress the information sequence into minimum possible bits (entropy level compressions) while channel encoding introduces redundancy to make the transmission data more robust. A tradeoff needs to be attained between source and channel coding in order to achieve both maximum compression of data bit-streams and optimum redundancy introduction for successful transmission of error resilient image/video sequences as discussed next.

2.2.2 Joint Source/Channel Coding

A common approach for building joint source/channel codecs [36] is to cascade an existing source codec with a channel codec wherein the key aspect lies in distribution of the source bits and channel bits between the source and channel codecs so that the resulting distortion is minimized.

With bandwidth being the only constraint, solution to the optimal source/channel bit distribution problem can be approached by first constructing the operational distortion rate curve as a function of bits for each sub-band of a wavelet decomposition and then applying one dimensional bit allocation algorithm. The optimal distribution of bits within a sub-band is done by using exhaustive search through all combinations of channel coding rates and quantization step sizes. One common thread among these analysis is that the joint source/channel codec is adaptive to the channel condition, which is assumed to have been estimated correctly.

The joint source/channel coding involves (a) finding an optimal rate allocation between source coding and channel coding for a given channel loss characteristics, (b) designing a source coding scheme, which includes the specification of the quantizer to achieve its target bit rate, and (c) designing/choosing the channel codecs to match the channel loss characteristics and achieve the required robustness.

Based on the feedback information system in the design of the joint source/channel coding scheme, the optimizer makes an optimal rate allocation between the source coding and channel coding mechanisms and conveys this information to the source and the channel encoders. The source encoder, then chooses an appropriate quantizer to achieve its target bit rate and the channel encoder chooses a suitable channel code to match the channel loss characteristics. This results in important low frequency sub-bands of images being shielded heavily using channel codes while higher frequencies are shielded lightly. This UEP technique reduces channel coding overhead, which otherwise is more pronounced in bursty wireless channels.

The QoS guarantees of wireless channel models are quite similar to those of differentiated service networks with a major difference being that the parameters involved rapidly change with time. This could be taken care of by allocating bandwidth to VBR in multiple ways that depend on channel variation with time. To this end, one way is to allocate the bandwidth equal to the actual packet rate of the video stream so that all the data can be delivered to the destination without any delay. When the application can tolerate large buffering delay at the source and in the network, the bandwidth requirement can be decreased. Apart from bandwidth allocation methods at the channel end, a performance measure yet to be determined, is desirable at the transmitter/receiver end to conduit these guarantees into a deployable model that fits well with the source video QoS requirements.

2.3 Error Concealment

Residual errors are inevitable when transmitting a video signal, regardless of the error resilience and channel coding methods used. Decoder error concealment refers to this recovery or estimation of lost information due to transmission/channel errors. Assuming a motion compensated video coder, there are three types of information that may need to be estimated in a damaged MB: the texture information, including the pixel or DCT coefficient values for either an original image block or a prediction error block; the motion information consisting of motion vectors for a MB coded in either P- or B- mode; and finally the coding mode of the MB. For coding mode information, the techniques used are driven by heuristics. The other two approaches are considered in the following subsections.

2.3.1 Recovery of Texture Information

All the techniques that have been developed for recovering texture information make use of the smoothness property of image/video signals and essentially they all perform the same kind of spatial/temporal interpolation. The MV field to the lesser extent, also shares the smoothness property and can be recovered by using spatial/temporal interpolation. The texture information recovery techniques include:

1. *Spatial Interpolation*: It is the technique of interpolating the damaged blocks from pixels in adjacent correctly received blocks. Usually, because all blocks or MBs in the same row are put in the same packet, the only available neighboring blocks are those in the current row and the row above, and typically boundary pixels of the neighboring blocks are used for interpolation purposes. Instead of interpolating the individual pixels, a simpler approach is to estimate the DC coefficient of the damaged block and replace the damaged block by a constant equal to the DC value which can then be estimated by averaging the DC values of the surrounding blocks. One way to facilitate such spatial interpolation is by an interleaved packetization mechanism so that the loss of one packet will damage only every other block. The missing DCT coefficients of the displaced frame difference are estimated by applying a maximal smoothness constraint at the border pixel of the missing block, where first and second order derivatives are used for quantifying smoothness. Since the DCT transformation is linear, the computation can also be performed in the pixel domain. Another spatial interpolation approach is to use the projection onto convex sets (POCS) technique. The general idea behind POCS based estimation method is to formulate each constraint about the unknown as a convex set. The optimal solution is the intersection of all the convex sets, which can be obtained by recursively projecting a previous solution onto individual convex sets.

2. *Temporal Interpolation*: MC temporal prediction is an effective approach to recover a damaged MB in the decoder by copying the corresponding MB in the previous decoded frame, based on the MV for this MB. The recovery performance by this

approach is critically dependent on the availability of the MV, which must be first estimated if it is also missing. MC temporal error concealment techniques might provide better results than any of the spatial interpolation techniques. To reduce the impact of errors in the estimated MVs, temporal prediction may be combined with spatial interpolation. MPEG-2 provides the capability of temporal error concealment for I-Pictures, since the transmission of additional error concealment motion vectors is allowed in MPEG-2.

A shortcoming with spatial interpolation approach is that it ignores the received DCT coefficients. This could be resolved by requiring the recovered pixels in a damaged block to be smoothly connected with its neighboring pixels both spatially in the same frame and temporally in the previous/following frames. If some of the DCT coefficients are received for this block, then the estimation should be such that the recovered block be as smooth as possible, subject to the constraint that the DCT on the recovered block would produce the same values for the received coefficients. The above objectives can be formulated as an unconstrained optimization problem and the solutions under different interpolation filter in the spatial, temporal and frequency domains.

2.3.2 Coding Modes and Motion Vectors

Inter frames are reconstructed using the motion vectors and the DCT coefficients of the prediction error and therefore, the loss of the motion vectors seriously degrades the decoded video. This degradation propagates to the subsequent inter frames [21] until an intra frame is encountered. In case of H.263, the loss of a MB motion vector propagates to the remaining MBs in the frame. In other standards including H.261, the previous motion vector is used for the encoding rather than the median of the neighboring vectors.

To facilitate decoder error concealment, as an added option, the encoder may perform data partition to pack the mode and MV information separate partition and transmit them with more error protection. This is the error resilient mode for both H.263 and MPEG-4. Even after this error resilience implementation, there is a fair chance of the mode and MV information being damaged. One way to estimate the coding mode for a damaged MB is by collecting the statistics of the coding mode pattern of the adjacent

MBs and find a most likely mode given the modes of surrounding MBs. An simple approach is to assume that the MB is coded in the intra-mode and use only spatial interpolation for recovering the underlying blocks.

For estimating the lost MVs, there are several operations: (a) The lost MVs can be assumed to be 0s, which performs well for video sequences with relatively small motion, (b) The MVs of the corresponding block may be used in the previous frame, (c) The average of the MVs from spatially adjacent blocks may be used, (d) The median of MVs from the spatially adjacent blocks can also be used, and (e) MVs could be re-estimated. It has been observed that the last two methods produced the best reconstruction results. Different MVs can be used for different pixel regions in the MB, instead of estimating one MV for a damaged MB to obtain a better result.

2.4 An End-to-End Perspective

A QoS guarantee is performed in each operational unit of a video communication system shown in Figure 2.1 and in all the error control mechanisms discussed above since it requires QoS guarantees to achieve its effective predetermined quality. Usually, these requirements [10] vary with respect to the application, perceptual quality, bandwidth and time. User related QoS is subjective and can be related in terms of spatial and temporal resolution scalability, SNR, and resolution scalability. The spatial resolution of the perceived video is a measure of the number of pixels in each frame while the temporal resolution is the number of received frames in unit time (frames per second, fps or sometimes also referred to as bit-rate in terms of bandwidth criteria). SNR resolution scalability is the allowable loss to the visual quality of the video and is realized by adjusting the degree of quantization during the video coding process. When a larger quantizer scale is applied, the quality of the decoded block reduces, which leads to degraded SNR values. However, the coded block size can become smaller, which has a positive effect from a view point of effective resource usage within the network.

As seen above, the QoS requirements of the source video are predominantly different from the QoS guarantees of the wireless channel/network described in Section 2.2, more so in case of real-time video communication where the QoS requirements of the application also change with time. This problem of relating these variations and bridging them is yet to be addressed. One way to consider this would be to statistically model the QoS requirements with variation in time and bandwidth of the channel and develop with a measure of the perceptual quality at the receiver end. An added constraint would be to force this measure to be a constant throughout the time and bandwidth variation for a reliable and robust real-time video communication. This approach will be discussed in Section 2.5. However, this measure is subjective and varies with varying applications and end users.

The problems with the existing solutions described in Sections 2.2 - 2.4 are that firstly, they are fragmented and not integrated. In each step of the video communication system in Figure 2.1, a local optimum is sought rather than a global one in that the efforts to minimize channel errors are independent at the transmitter and receiver ends. Secondly, these efforts are open ended giving way to unnecessary overhead when each of the techniques considers its best optimum performance. In such a scenario, the performance of the integrated system is jeopardized to match the independent individual unit's optimum. And lastly, these techniques are non-adaptive with regard to bandwidth, display and viewing conditions, and application types.

The above described problems in existing approaches lead way to an integrated adaptive approach which requires an end-to-end QoS criteria based on the visual quality. This approach aims at having maximum possible perceptual quality at the viewers' ends with a pre-determined latency and a specified channel bandwidth. The QoS measure would then suffice the visual quality requirements for a given latency at a cost of required channel bandwidth, power consumption and algorithm complexity. In effect, the wireless video codec should aim at achieving a constant perceptual quality (CPQ) by using an integrated end-to-end QoS oriented performance criteria.

2.5 Adaptive Constrained UEP Scheme

It is maintained here that the fraction of source bits that need to be transmitted to achieve an acceptable level of perceptual quality is a function of the video source content. Abrupt scene changes and irregular motion require higher number of bits compared to stationary/gradually changing scenes. Hence, an adaptive UEP scheme is required that dynamically optimizes the UEP protection levels in a packet based on the content of the scene as well as the channel conditions.

In this section, an adaptive constrained optimization is implemented by dynamically varying the UEP level in a packet based on the channel conditions. The transmission is started with a predetermined UEP level. A "forcing function" is then estimated based on the feedback of channel loss characteristics which provide the bandwidth and the latency constraints. The UEP level in the packet is then modified each time the channel is estimated in accordance with this forcing function, which is defined and discussed in the subsequent sections. The UEP level is chosen such that the visual quality is maximized under the given bandwidth and the latency constraints.

Similar approaches have been evaluated in [49], [50] and [51]. While Mohr *et al.* [50] uses unconstrained optimization, a constraint on system probability of failure rather than on channel conditions is applied in [49]. In doing so, Grangetto *et al.* have enforced a tight bound on minimum achievable Peak signal-to-noise-ratio (PSNR) but have not prevented any channel inflicted quality loss. It is argued here that channel constraints have to be implemented adaptively and the packet structure changed dynamically for the model in [49] to work effectively. In [51] however, an unconstrained optimization is carried out on the overall rate distortion performance of a joint source-channel coding system.

2.5.1 Problem Formulation

The problem is formulated as a constrained optimization of UEP level in the packet structure to achieve maximum expected subjective perceptual quality. Objective measures that accurately define subjective visual quality include signal-to-noise-ratio (SNR) measurement, measures that use frequency domain masking and pooling, and detection thresholds. Moore *et al.* presented a well designed measurement of detection thresholds for video sequences with artifacts in [52]. In this work, however, *PSNR* is adopted. The problem statement can be defined as finding an optimum UEP level that maximizes the end user PSNR first by using an unconstrained optimization under the conditions explained in [11] for effective bandwidth and effective channel capacities and then constraining it using variable bandwidth and latency constraints. The notion of effective bandwidth is also well described with regard to allocation in [53] and with regard to multi-users in [54].

The optimization of UEP is done based on a *forcing function*, which can be defined as the function that determines the error protection level to be employed in a packet for the given channel conditions, i.e., for a given combination of effective bandwidth and loss characteristics, the forcing function evaluates the error protection bits required in the packet to be transmitted. It therefore indirectly "forces" a UEP scheme for the packet transmission based on current channel conditions.

Let this forcing function be represented by $f(\mathbf{A}_i)$ where $\mathbf{A}_i$ is a column vector representing the i-th packet sent. Mathematically, first the unconstrained problem can be defined as follows: Let μ be the UEP ratio (protection level) in the packet. Then, for a given $f(\mathbf{A}_i)$, we need to find the μ that satisfies

$$\max_{i \in [0,N]} \{PSNR/f(\mathbf{A}_i) \quad \forall \mathbf{A}_i\}, \tag{2.1}$$

where N is the total number of packets. For simplicity, we assume $f(\mathbf{A_i})$ to be the performance curve that includes the maximum area on the probability of successful arrival of the packet. It can be noted that Eq. (2.1) poses a considerable challenge to solve even without the bandwidth and the latency constraints because of the time varying nature of the forcing function.

The system performance of a standard wireless network, cdma2000, is observed for varying bandwidths over time for the consideration of constrained optimization problem. Once the transmission rate (R) matches the bandwidth (B) within the Acceptable Range (AR), the rate is either fixed or lowered. Similarly, bounds on latency are also applied. For this, the upper bound of latency is considered to be the time difference (T) between two I-frames in the video transmission. Let t_1 be the time required to transmit the entire frame with UEP ratio of μ. It is given by

$$t_1 = \frac{N \times S \times (1-\mu)}{B + AR}, \tag{2.2}$$

where S is the packet size, N is the total number of packets, B is the bandwidth, and

$\mu \in [0, 1]$. The latency constraint is then given by

$$|t_1| < t_0 + T, \tag{2.3}$$

where t_0 is the time required to transmit the entire frame without any delay bounds.

The bandwidth constraint indirectly limits the total number of packets transmitted for a given t_1 in Eq. (2.2). Hence, even though a total number of N packets are required to transmit the entire video frame, a total of say τ packets can only be transmitted by a reduced bandwidth in the given time constraint in Eq. (2.3). This is particularly true for higher loss-protected transmissions where fewer source bits are sent in each packet. Hence, the bandwidth constraint imposes indirectly a reduction threshold on the total number of packets given as

$$\tau = N\chi = N\frac{\mu_1}{\mu_i}, \tag{2.4}$$

where μ_1 is the UEP ratio that satisfies t_1 in Eq. (2.2) and μ_i is the UEP ratio for the current packet transmission. Here, χ is called the *UEP loss factor*. The variation in bandwidth can have considerable effect on the quality of the video transmitted. The end user visual quality acceptance variation [52] gives us the acceptable range (β) of bandwidths to achieve a constant perceptual quality. This implies that the rate of transmission should adapt to these bandwidth variations with an allowable range of β, which gives us the bandwidth constraint as

$$|R - B| \leq AR. \tag{2.5}$$

Hence, the constrained optimization problem can now be formulated as: To find a μ that satisfies

$$\begin{aligned} \max_{i \in [0,k]} \quad & \{PSNR/f(\mathbf{A}_i) \quad \forall \mathbf{A}_i\}, \\ s.t: \quad & \\ & |t_1| < t_0 + T, \\ & |R - B| \leq AR, \end{aligned} \tag{2.6}$$

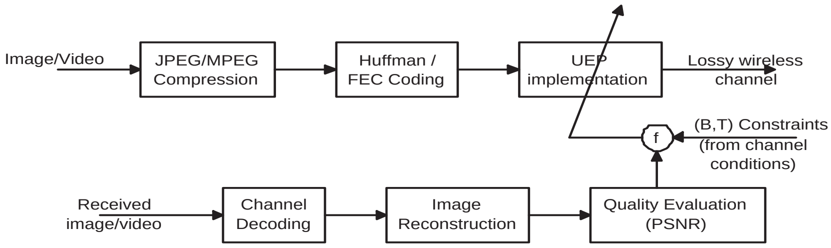


Figure 2.2: Structure of the proposed algorithm

where $k = min\{N, \tau\}$.

2.5.2 Adaptive UEP Optimization

As the forcing function is time-varying, each packet $\mathbf{A}_i$ needs to be considered independently for a fixed $f(\mathbf{A}_i)$ to obtain a UEP level μ that maximizes the PSNR. Since the packets are considered independent of one other, a dynamically adaptive algorithm is developed in which the UEP level changes from packet to packet with changing network conditions so that the end user perceptual quality is always maximized.

The block diagram in Fig. 2.2 describes the basic structure and operation of the proposed algorithm. The UEP ratio is then decided and/or varied by considering the end user visual quality and channel constraints. After the UEP part of the header is appended to the packets, the wireless channel packet losses are simulated.

Constrained Optimization

As the bandwidth B, the latency time t_0, and the acceptable range of bandwidth β are time dependent, they are denoted by $B(t)$, $t_0(t)$ and $\beta(t)$, respectively. Therefore, the per-packet error protection in terms of the forcing function can now be written as

$$\mu_i = f(B(t), t_0(t), \beta(t)). \tag{2.7}$$

For solving the optimization problem, we first need to express the constraint as well as the arguments of the optimization as a function of the packet UEP parameter μ_i. Since N and S are positive, and recalling the latency constraint in Eq. (2.3) and combining that with Eq. (2.2), we obtain

$$|1-\mu_1| < \frac{(t_0(t)+T)\,|B(t)+\beta(t)|}{NS}, \tag{2.8}$$

where μ_1 is as defined in Eq. (2.4). Since μ_1 cannot be greater than 1, $1-\mu_1$ is positive. Therefore, we have

$$\mu_1 > \left|1-\frac{(t_0(t)+T)|B(t)+\beta(t)|}{NS}\right|. \tag{2.9}$$

Now, consider the bandwidth constraint in Eq. (2.5). The time varying equation can be written as $|R-B(t)| \leq \beta(t)$. By rate-distortion theory, there is a limit on the rate at which we source code the data such that the distortion is kept under the threshold we want. This rate (in terms of protection) is given by

$$R \leq B(t)\left[1-\frac{\mu_1}{\mu_i}.\right] \tag{2.10}$$

The equation comes from considering the bandwidth constraint with the channel capacity. Note that here, we consider the "effective" bandwidth of the channel as defined in [11]. Therefore, from Eq. (2.10) and the time modified version of Eq. (2.5) we have

$$|R-B(t)| \leq \left|B(t)\frac{\mu_1}{\mu_i}\right| \leq \beta(t) \implies \mu_i \geq \left|\frac{B(t)\mu_1}{\beta(t)}\right|. \tag{2.11}$$

Substituting the expression for μ_1 from Eq. (2.9), we have

$$\mu_i > \left|\frac{B(t)}{\beta(t)}\left[1-\frac{(t_0(t)+T)|B(t)+\beta(t)|}{NS}\right]\right|. \tag{2.12}$$

We can further simplify the constraint by means of certain assumptions (and find a good closed form expression for it). If here, we assume synchronization of the channel error estimation and packet transmission intervals, i.e., if the channel estimation at the receiver is done at the same time instances as the packet arrival times, then $B(t)$, $t_0(t)$ and $\beta(t)$ can be replaced with B_i, t_{0i} and β_i, respectively. Based on this assumption, the constraint can be further simplified as

$$\mu_i > \left|\frac{B_i}{\beta_i}\left[1-\frac{(t_{0i}+T)|B_i+\beta_i|}{NS}\right]\right| = \left|\frac{B_i}{\beta_i}-\frac{t_{0i}B_i^2}{NS\beta_i}-\frac{t_{0i}B_i}{NS}-\frac{TB_i^2}{NS\beta_i}-\frac{TB_i}{NS}\right|. \tag{2.13}$$

Note here that the value of $B_i + \beta_i$ is considered positive. This also aids an assumption later made regarding the variation of β_i with reference to B_i.

Consider t_{0i} in the above equation. It can be considered as an average time for receiving all packets in a frame and can be assumed to be much higher than the latency produced on a per packet basis.[2] Under this assumption $t_{0i} \approx t_0$, a constant over the N packets.

Since T, N, and S are also assumed to be constant, we can define a constant term C as

$$C = \frac{t_0 + T}{NS} \;\; = \text{ aconstant.} \tag{2.14}$$

Using this definition, Eq. (2.13) can be rewritten as

$$\mu_i > \left| \frac{B_i}{\beta_i}(1 - C\,|B_i + \beta_i|) \right|. \tag{2.15}$$

When PSNR is used as a quality measure, since $\beta_i < B_i$ is always true (proved mathematically later on by Eq. (2.19)), let us define γ_i to be the acceptable bandwidth ratio given by

$$\gamma_i = \frac{\beta_i}{B_i}. \tag{2.16}$$

Ideally, we would want γ_i to be 0 since any variation in the acceptable bandwidth β_i would disturb the "constantness" of the received video quality. It is obvious that the affect on perceptual quality of the video due to this variation in β_i is dependent on B_i a little differently than PSNR. The higher the value of B_i, the lower the effect of β_i variation on the perceptual quality. It will be discussed later how we can use this variation to our advantage.

By substituting Eq. (2.16) in Eq. (2.15), we have

$$\mu_i > \left| \frac{1 - CB_i(1 + \gamma_i)}{\gamma_i} \right| \; = \left| \frac{1 - C(B_i + \beta_i)}{\gamma_i} \right|. \tag{2.17}$$

Note that the protection is not directly dependent on the latency constraint but the

[2] It should be noted here that the loss probability in each frame is *not* assumed to be constant. This assumption merely states that the time to receive the entire frame (even with time varying losses) is almost a constant. This can be extended to a video as t_{0i} being the time average of all the frames to be transmitted.

fact that B_i forces N to tend towards k (from Eq. (2.4) and Eq. (2.6)) implies that we only transmit k packets in the allocated time (given by the latency constraint in Eq. (2.3)). Since we want $\gamma_i \to 0$, to maximize the constant quality at the end user, we want as much protection to the transmitted packets as possible. This can be seen from Eq. (2.16) and Eq. (2.17). When $\gamma_i \to 0$, $\mu_i \to \infty$. But for solving Eq. (2.6), we face the following problem with k as seen from Eq. (2.4). $k = min\{N, \tau\} \;=\; min\{N, N\chi\} \;=\; min\{N, N\frac{\mu_1}{\mu_i}\}$. Therefore, when $\gamma_i \to 0$, $\mu_i \to \infty$ and so $k \to 0$. Recall that here k is the number of packets required to transmit the entire frame (cut down from N due to latency and bandwidth constraints). And when k tends to 0, the PSNR is reduced drastically instead of increasing with protection (since μ_1 also $\to 0$ when $\gamma_i \to 0$, the drop in k and so the drop in PSNR is much faster). For this reason, an optimum value of μ_i is necessary.

Before we proceed to find this optimum, let us see if we can extract any information from what little variation we have with varying the acceptable range of bandwidth. This information can be obtained by finding the bounds of γ_i and exploiting its variation within these bounds. Practically, the encoded bit rate is not always equal to or even close to the available bandwidth, which implies that $\beta_i \neq 0$. In fact, it is a small value greater than 0. This can also be seen by setting $R \leq B_i$ in Eq. (2.5). Therefore, we can safely set the upper bound of γ_i to be 1.

The lower bound of γ_i can be found by considering the upper limit on the number of transmitted packets. We need to decrease γ_i so that k is as close to N as possible. If we assume a strict bandwidth constraint, we have τ always less than N. Let τ_{min} be the value of τ that satisfies this strict constraint. Since $k = min\{N, \tau\}$, $k = \tau_{min}$, and so from Eq. (2.17)

$$k = N\chi_{min} \;=\; N\frac{\mu_{1min}}{\mu_i} < N\left[\frac{1 - C\,|B(t) + \beta(t)|}{\left\{\frac{1-C(B_i+\beta_i)}{\gamma_i}\right\}}\right]. \tag{2.18}$$

Under the synchronization assumption, we have $k < N\gamma_i$. Hence, the limits of the acceptable bandwidth ratio are

$$0 < \frac{k}{N} < \gamma_i < 1. \tag{2.19}$$

Using these limits, at the receiver, we decide the apt β_i that gives a good PSNR performance at the receiver and send it to the transmitter using a feedback based network. At the transmitter, we vary the source rate R such that the required μ_i is obtained for maintaining the PSNR and also the constraints are practically made viable.

Next, we need to find the expression for PSNR as a function of the packet protection ratio μ_i. For doing this, we first need to express the PSNR as a composite of per-packet estimated PSNR values (let us call them $PSNR_i$). Using the definition of PSNR, the per-packet PSNR for the i-th packet can be defined as

$$PSNR_i = \frac{255^2}{\sum\limits_{(p,q)\in[0,i]}\sum [f(p,q) - r(p,q)]^2}, \tag{2.20}$$

where $f(p,q)$ is the original image or video frame at (p,q)-th pixel location and $r(p,q)$ is the received video frame at that pixel location.[3] The effective expected PSNR to be maximized, $E\{PSNR\}$, can now be written in terms of individual per-packet PSNRs as

$$E\{PSNR\} = E\{PSNR_N\} = E\left\{\frac{1}{\delta}\sum_i PSNR_i\right\} = \frac{1}{\delta}\sum_i E\{PSNR_i\}, \tag{2.21}$$

where δ is a scale factor that removes the unwanted repetitions of the per-packet PSNR summation. Typically δ will have expressions of the form $(N-1)![(N-1)!+1]/2$. By the total probability theorem, $E\{PSNR_i\}$ can be further expanded as conditional probabilities given by

$$E\{PSNR_i\} = \sum_p \sum_q E\{PSNR_i/(p,q) \in \mathbf{A}_i\}\, Pr\{(p,q) \in \mathbf{A}_i\}. \tag{2.22}$$

We can observe that the probability of $(p,q) \in \mathbf{A}_i$ is related to the error protection given to $\mathbf{A}_i$ and can be expressed as

$$Pr\{(p,q) \in \mathbf{A}_i\} = \frac{(1-l_i)(1-\mu_i)}{N}, \tag{2.23}$$

where l_i is the loss probability of the packet $\mathbf{A}_i$. The first term in the right hand side of

[3] Note that the limits of the summation $(p,q) \in [0,i]$ define the effective PSNR *till* the i-th packet (and not just that of the i-th packet). PSNR$_N$ therefore, would give us a measure of the actual PSNR.

Eq. (2.22) can be further simplified as

$$\begin{aligned} E\{PSNR_i/(p,q) \in \mathbf{A}_i\} &= E\left\{ \frac{255^2}{\sum\sum_{(p,q)\in[0,i]} [f(p,q) - r(p,q)]^2} / (p,q) \in \mathbf{A}_i \right\} \\ &= \frac{255^2}{\sum\sum_{(p,q)\in[0,i]} [f(p,q) - r(p,q)]^2} \left(\frac{1-\mu_i}{k} \right). \end{aligned} \tag{2.24}$$

Now, substituting Eqs. (2.23) and (2.24) in Eq. (2.22), we get the expression for the per-packet PSNR in terms of the error protection given to each packet. Before we find the expression for the argument in the optimization, for the sake of simplicity, we define the pixel wise dependence of PSNR with a new variable $\hbar_i$. This variable depends on the each packet (for optimization purposes) and can be defined as

$$\hbar_i = \frac{1}{\delta} \sum_p \sum_q \left[\frac{255^2}{\sum\sum_{(p,q)\in[0,i]} [f(p,q) - r(p,q)]^2} \right]. \tag{2.25}$$

Using equations Eqs. (2.22) - (2.25) and back substituting, we can further simplify Eq. (2.21) as

$$E\{PSNR\} = E\{PSNR_N\} = \sum_i \frac{\hbar_i (1-l_i)(1-\mu_i)^2}{kN}. \tag{2.26}$$

This expression is important because of two reasons. Firstly, it gives us a good idea about the parameters on which the objective quality depends upon based on a per-packet transmission and secondly, it expresses the quality measure at the end user in terms of the protection given to each packet. Therefore, we can now express the constrained optimization problem as

$$\begin{aligned} &\max_{\mu_i} \quad \tfrac{\hbar_i(1-l_i)(1-\mu_i)^2}{kN}, \\ &s.t: \\ &\qquad \mu_i > \left| \tfrac{1-C(B_i+\beta_i)}{\gamma_i} \right|, \quad \mu_i < 1, \\ &\qquad \tfrac{k}{N} < \gamma_i < 1. \end{aligned} \tag{2.27}$$

It is now sufficient to solve this expression for μ_i. As seen from the expression, this is in the form of a standard non-linear optimization problem. Any of the conventional methods such as the reduced-gradients, sequential linear and quadratic programming methods, or methods based on augmented Lagrangian and exact penalty functions can be followed to solve Eq. (2.27). The only deviation to this problem comes in the form of non-convexity on the bounds of the constraints.

Greedy algorithm

Multiple assumptions have been made to represent Eq. (2.6) in the form of Eq. (2.27). The validation of these assumptions cannot be done mathematically. However, one way of solving Eq. (2.6) non-mathematically is by setting an approximate range for each of the variables and finding the location of the optimum μ. Another way is to employ a greedy algorithm that finds the optimum packet protection such that the losses are minimized. There have been algorithms proposed for such rate-distortion optimizations but none of them operate on a per-packet basis. However, these algorithms can be used to represent and solve the distortion minimization problem on a per-packet basis.

For obtaining the argument of this optimization, let α_i be the protection given in bits to the packet. Let us also assume that the criterion for minimization be certain L-norm of the packet $\mathbf{A}_i$ which depends on the packet protection α_i. Currently, the size of the packet, S, is assumed to be fixed. However, an extension to this optimization can be considered where the packet size is varied. The greedy technique to find the minimum μ_i can then be stated as

$$\begin{aligned}
&\min_{\alpha_i} \quad \sum_i l_i(1-\alpha_i)\left\|\mathbf{A}_i\right\|_L, \\
&s.t: \\
&\qquad \alpha_i > S\left|\tfrac{1-C(B_i+\beta_i)}{\gamma_i}\right|, \quad \alpha_i < S, \\
&\qquad \sum_i |\alpha_i| \le mS, \qquad m \ll k,
\end{aligned} \tag{2.28}$$

where m is a positive integer.[4] Note here that the channel constraints remain the same

[4] An important note to be made from Eq. (2.28) is that the effective argument is less than what we considered for Eq. (2.27). Here, the minimization is not based on the PSNR anymore due to the fact that variation of the quality to the higher side is redundant in terms of PSNR. It is rather sensible to consider

as before and so, the constraint in Eq. (2.27) can be directly applied in terms of α_i. The second constraints merely states that the protection applied is not large compared to the amount of data transmitted. In other words, this constraint bases itself on the assumption that the protection α_i given to the packet does not change the norm of the packet. If this important assumption is not considered, the problem would change very dynamically and so the formulation of the constraints needs to be modified each time the packet is transmitted. This in turn is not only very complex to solve but also inappropriate due to the bursty nature of the channel.

The solution to Eq. (2.27) or Eq. (2.28) gives the upper bound on the per-packet protection ratio μ_i. For maintaining constant perceptual quality, we have to define the minimum UEP ratio μ_i such that the effective quality of the received video does not vary much on the higher side. This upper limit on the quality is usually decided at the receiver when the first or the first few frames of the video are received. The transmitter then changes the packet protection based on the feedback it receives from the receiver regarding the lower limit on μ_i. The optimization for the upper bound on quality can also be done at the receiver by introducing losses/noise or other degradations.

2.5.3 Simulation Results

To account for the channel effects, consider $\mathbf{A}_i$ to be a column vector of size $S \times 1$. Let $\mathbf{D}$ be the frame vector formed by lexicographic ordering of all the $\mathbf{A}_i$s. Therefore, $\mathbf{D}$ is $[\mathbf{A}_1^T \mathbf{A}_2^T ... \mathbf{A}_k^T]$, where k is as defined in Eq. (2.6). The size of $\mathbf{D}$ is $1 \times Sk$ bits or $1 \times k$ packets. $\mathbf{C}$, the channel output, can then be obtained as $diag(\mathbf{P^T D})$, where $\mathbf{P}$ is the binary probability loss vector of the channel with a predefined loss percentage. $\mathbf{P}$ is a row vector of size $1 \times k$ and is randomly generated. Since each element of $\mathbf{P}$ is multiplied with each $\mathbf{A}_i$ in $\mathbf{D}$, $\mathbf{C}$ is a vector of k packets and contains the received set of packets which are decoded from which the image is reconstructed.

The compressed output stream is vectorized and multiplied with the vector $\mathbf{P}$ that

the effective overall loss induced and the amount of redundant protection required to keep the upper limit of quality from crossing the upper bound of acceptable range (keeping in mind that any higher quality, even though is good for viewing, will produce a variation that would be visible and when the bandwidth reduces at a later point of time, this variation would introduce perceivable annoyance). In terms of PSNR, this would be similar to considering $E\{PSNR\} - \sum_i Var_i^2$, where Var_i is the per-packet loss variance.

Table 2.1: Optimum overall μ values and the corresponding PSNR values (in dB) for a mean loss probability = 0.15 and loss variance = 2.5%

	S = 128		*S = 256*		*S = 512*	
Image	μ_{avg}	PSNR	μ_{avg}	PSNR	μ_{avg}	PSNR
Lena	0.1736	28.56	0.1804	28.83	0.1769	27.91
Cameraman	0.1823	29.61	0.1830	29.76	0.1828	29.37
Barbara	0.1941	26.13	0.1954	26.47	0.1930	26.22
Brain	0.1737	31.42	0.1801	31.65	0.1743	30.99
Sail	0.1684	29.14	0.1702	19.58	0.1653	29.22
Football	0.1848	28.86	0.1859	29.19	0.1861	28.82
Hockey	0.1851	29.93	0.1913	30.45	0.1889	30.06
Yogi	0.1892	28.63	0.1926	28.91	0.1971	28.11

is randomly generated using Monte Carlo simulation. About 1000 varying packet loss simulations have been generated independently for each transmission and their statistical average has been taken to obtain the probability loss vector. Here, each packet is considered to have a uniform loss distribution, thus making the loss distribution of the entire frame to be Gaussian.

Table 2.1 encapsulates the performance of the proposed algorithm. Eight images were considered as independent video segments. For each image, an optimum μ (the average of all the individual packet μ_is) was found by experimentation for a fixed loss percentage and the corresponding maximum PSNR was noted. The loss percentages were then varied and each time a new μ, optimum for that particular loss percentage, was recorded. This experiment was repeated for varying packet sizes while keeping the number of packets well within the constraints given in Eq. (2.6).

A sample curve of PSNR variation with UEP ratio (μ) in packets of size $S = 512$ bits for fixed mean loss of 0.15 percent and variance of 2.5 is shown in Fig. 2.3. The curve gives us the maximum PSNR value for a fixed loss percentage and the optimum μ for which this PSNR is attained.

The variation of maximum achievable PSNR with varying packet loss percentages for a fixed packet size, $S = 512$, can be seen in Fig. 2.4(a). The variation of all five images are plotted for comparison. Also shown is the performance of the system with a constant UEP fixed at $\mu = 0.33$ (the dotted line) for the cameraman image. As seen from the figure, the dynamically adaptive algorithm clearly outperforms the constant

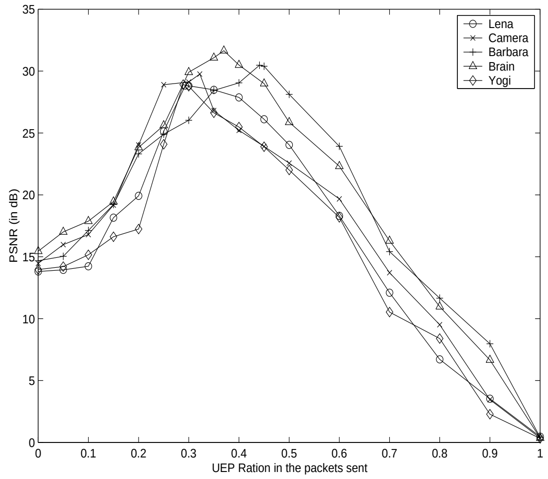
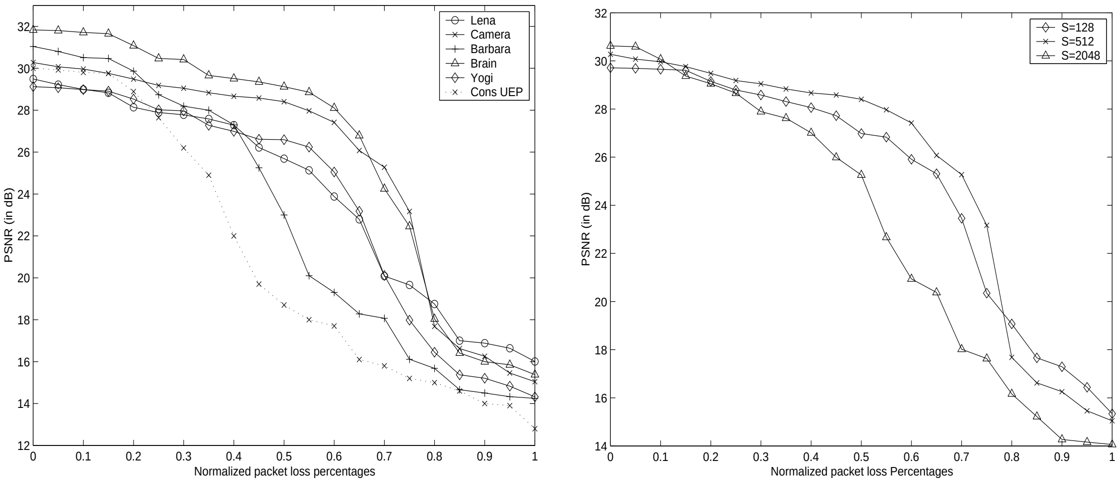


Figure 2.3: Sample curve of PSNR variation with UEP level (μ) in the packet for a fixed mean loss probability of 0.15 percent with variance 2.5%, $S = 512$.

UEP technique.

Also seen from Fig. 2.4(a) is the fact that the objective quality measure, PSNR, drops rapidly after a particular loss probability. For example, the PSNR values of the cameraman image (for packet size $S = 512$) dropped from 25.27 for $l = 0.7$ to 17.68 for $l = 0.8$. An implementation aspect here would be to restrain from transmitting the video frames if the channel loss probability crosses a particular *threshold* on or after which the PSNR of the received video drops radically making it difficult to maintain high perceptual quality. Here, $l = 0.7$ can be considered as the *threshold* for the cameraman image.

Fig. 2.4(b) shows the variation of PSNR for the cameraman image for varying packet sizes. It is observed that the performance of the system first improves with increasing packet sizes, especially for higher packet loss percentages, and then decreases for particularly large packet sizes. This implies that an optimum value of S needs to be chosen to obtain higher PSNR for similar channel conditions. It can also be seen from the figure that a single S is not optimum for all packet loss probabilities. At lower loss percentages, a large S is optimal, whereas, for higher packet losses, a smaller S would give better results. Hence, to achieve maximum perceptual quality at the end user, it can be deduced that a dynamic variation of S is also necessary.



(a) (b)

Figure 2.4: (a) PSNR variation with varying normalized packet loss percentages (l); $S = 512$, and (b) PSNR variation with varying packet sizes (S) for the *Cameraman* image.

2.6 Summary

This chapter reviewed the most recent advancements in adaptive and interactive wireless video communications with primary focus on the error control tools and mechanisms of different video coding standards. Due to the effects of noisy fading wireless channel, it was found that error resilience and error concealment were the most important aspects of current research for successful realization of transmission and reception of image/video signals over bandwidth limited fading wireless networks/channels. The chapter also provided an overview of the source and channel encoding implementations at the transmission end and specified the significance of the encoder error resilience. The channel characteristics and its effects with regard to QoS requirements have been studied and a review of the differences between the QoS of video and channel is presented. The chapter discussed different error concealment techniques that can be implemented in the present day standards to take care of the channel/transmission errors in the obtained signal at the receiving end to obtain a near-original reconstruction of the transmitted information.

A dynamically adaptive constrained UEP technique is then proposed for wireless video transmissions that aims at achieving maximum perceptual quality at the end user.

The existing challenge is formulated as a constrained optimization problem. An algorithm for this technique is given where in the UEP is varied dynamically based on the constraints developed by varying wireless channel conditions. Simulation results of this algorithm on various video frames are presented and analyzed for varying loss conditions and packet structures. It can be seen from these results that the algorithm outperforms other currently existing techniques for transmitting video over wireless channels especially for higher packet loss percentages. Illustrative experiments are conducted to demonstrate the practical strategy of implementing such a UEP scheme that performs off-line optimization for different types of video segments and different channel conditions. The potential advantage of this approach is that it attempts to achieve maximum perceptual quality for the given channel conditions and gives better results for higher loss percentages at relatively low cost during run time. However, a possible disadvantage may lie in the fact that the transmitter requires an adaptive feedback for determining the optimal UEP. Making the transmitter aware of the current channel conditions without a feedback from the receiver is still to be researched.

Chapter 3

Watermark-based Error Concealment (WEC)

The transmission of images and video over wired and/or wireless channels introduces multiple losses into the transmitted data that manifest themselves as various types of artifacts. These artifacts degrade the quality of the received image/video as they vary rapidly during the course of transmission based on the channel conditions. Therefore, there is a need for a good error concealment technique that can detect and correct (or conceal) these errors better and display a good quality image/video regardless of the channel conditions [1]. In the case of video transmission over wireless channels, adaptive error control that adapts to the approaches both at the transmitter and at the receiver has proven to be more effective [23].

Error concealment methods use spatial and temporal information to recognize that an error has occurred. Once an error is detected, the received video stream is adjusted with an attempt to recover the original data. A number of error concealment techniques have been proposed in the literature that use either statistical methods to detect and correct errors (these are usually computationally intensive) or depend on certain critical information from the transmitter, like the re-synchronization markers, to detect these transmission errors.

In this chapter, we propose a set of novel watermark-based error concealment (WEC) algorithms. Watermarking is usually used to introduce some redundancy to the transmitted data with little increase in its bit rate during implementation [57].

The basic idea of a proposed WEC algorithm is as follows. A frame of a video is wavelet transformed and the low-low approximation coefficients (usually second or third order) are then embedded in the frame itself during MPEG encoding. The embedded data can then be error protected unequally such that the mark signal embedded packets are given higher protection against channel errors. At the receiver, the mark is extracted from the decoded frame. The channel corrupted information of the frame is then reconstructed using the embedded mark signal. Specific areas lost through transmission are selected from the reconstructed mark signal and replaced in the original frame, thus enhancing its perceptual quality [58]. We provide modified algorithms for implementation in high detail color extensions in the case of wired and wireless transmission scenarios. Also, a comparison of the proposed technique to its two-part variant (where the low resolution child image is encoded and transmitted as side information) is provided along with an extensive analysis of its performance.

3.1 Previous Work

The foundations for the use of data hiding as an error control tool were laid by Liu and Li [59]. They extracted the important information in an image, like the DC components of each 8×8 block, and embedded it into the host image. In the work that followed, Liu and Li's work formed the basis. Certain key features were extracted from the image and these features were encoded and data hidden in the original image either as a resilience tool or for concealment [60]-[62].

Watermarking of error correcting codes was introduced by Lee and Won [63]. Here, the parities generated by conventional error control codes were used for watermarking sequence. A region of interest (ROI) based coded bit stream embedding was employed by Wang and Ji, where the ROI DCT bit stream is embedded into the region-of-background wavelet coefficients [64]. This technique gives better results when perception based encoding is employed.

The concept was extended to video coding by Bartolini *et al.* [65]. However, they used data hiding as a tool to increase the syntax-based error detection rate in H.263 but not for recovering or correcting lost data. Munadi *et al.* extended the concept of key feature extraction and embedding to inter-frame coding [66]. In their scheme, the most important feature is embedded into the prediction error of the current frame. However, the effects on motion vectors and the loss of motion compensated errors were not addressed. Yilmaz and Alatan proposed embedding a combination of edge oriented information, block bit-length, and parity bits in intra-frames [67]. They use a minimally robust technique of even-odd signalling of DCT coefficients for embedding.

The problems with existing techniques are: (1) Only one or a few selected set of key features are used for embedding. These features may not necessarily follow the loss characteristics of the channel employed. (2) They use transform domain to encode the data that needs to be embedded, often DCT. However, if losses occur on the DC coefficient or a set of first few AC coefficients, the loss to the extracted reference would be significant and therefore may lead to reduction in concealment performance. (3) Almost all the techniques use fragile or semi-fragile data embedding schemes which are more susceptible to attacks. Our proposed technique avoids two of the three problems by embedding half-toned version of the whole reference image (instead of encoding its transform coefficients). This way, loss or errors in the data will have smaller and local effects on the reconstructed video. A possible solution to the third problem will be addressed in a future work.

A set of concealment techniques that do not use data hiding while giving similar high levels of performance has been proposed in the literature. Block based deterministic interpolation models are used for reconstruction of missing blocks in either the spatial domain [68]-[70] (simplified edge extraction imposition for obtaining the directional interpolation was considered in [68] and [69] while projection onto convex sets was considered in [70]) or spectral domain [71] (where lost DCT coefficients are estimated based on spatial smoothing constraints). Li and Orchard provided a good review of these techniques and proposed a set of block-based sequential recovery techniques [72]. These work well in simplified loss scenarios where successfully received data is assumed to be reconstructed loss free. This is often not the case. A comparison of these techniques with the proposed technique is provided in Section 3.5.

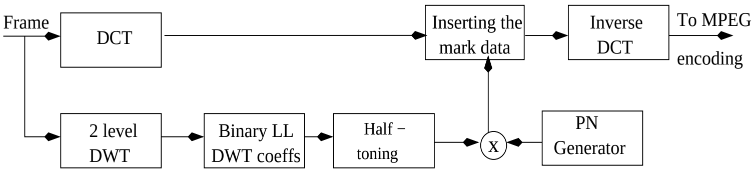


Figure 3.1: Block diagram of the embedding algorithm.

3.2 Watermark-based Error Concealment (WEC)

The proposed WEC scheme can be divided into an embedding part and a retrieval part. It should be noted that the proposed technique does not overload the communication channel by requiring feedback or any retransmission of damaged blocks.

3.2.1 The Embedding Part

The data hiding technique used here is a modified version of Cox's watermarking algorithm [73]. Due to the limited embedding capacity of the algorithm, it is practically not feasible to embed the whole frame (full resolution) into itself [74]. In this work, therefore, the discrete wavelet transform (DWT) and dithering techniques have been used to reduce the amount of data to be embedded such that the algorithm embeds maximum information while still catering to the feasibility issues.

Wavelets have several properties that make them good candidates for this application. Some of the important ones relevant to this algorithm are: (1) The approximation coefficients provide a good low-resolution estimate of the image, while minimizing the aliasing artifacts resulting from the reduction in resolution, and (2) The wavelet coefficients are localized, such that a corruption of a coefficient through channel errors has only local effect on the image. Dithering techniques make it possible to generate binary images which look very similar to the parent gray level images. The technique employed here is Floyd-Steinberg error diffusion dithering algorithm [75], [76]. The DWT approximation coefficients are half-toned before being embedded.

The block diagram of the embedding algorithm is shown in Fig. 3.1. The operation of the embedding part can be described as follows. The 2-D DWT of the frame is first computed. A second level DWT is performed again on the approximation coefficients to obtain an image that is $\frac{1}{16}$-th the size of the original frame. A half-toned image, the marker, is then generated from the reduced size image. One marker is used for each frame. After the marker is generated, each pixel of the marker is repeated 4 times in a 2×2 matrix format. This repetition operation allows the decoder to recover the marker from the data in a more robust fashion.

Mathematically, the reduced size image generated for the i-th frame, $\mathbf{f}_i$, can be represented as $\mathbf{m}_i$. Here, $\mathbf{f}_i$ is of size $m \times n$ and $\mathbf{m}_i$ is of size $\frac{m}{4} \times \frac{n}{4}$. The half-toning operation is performed on $\mathbf{m}_i$ using a Floyd-Steinberg diffusion kernel $\mathbf{D}_{FS}$ given by

$$\mathbf{D}_{FS} = \frac{1}{16} \begin{bmatrix} 0 & 0 & 0 \\ 0 & P & 7 \\ 3 & 5 & 1 \end{bmatrix}, \tag{3.1}$$

where P is the current pixel position. $\mathbf{D}_{FS}$ is typically applied on each 3×3 block of the reduced size image. The resulting marker is denoted as $\mathbf{w}_i$. Each pixel of $\mathbf{w}_i$ is then repeated in a 2×2 matrix format to form $\breve{\mathbf{w}}_i$. Note that $\breve{\mathbf{w}}_i$ is of size $\frac{m}{2} \times \frac{n}{2}$.

A zero mean, unit variance pseudo-noise image is then randomly generated with a Gaussian distribution and a known seed. A unique pseudo-noise image, $\mathbf{p}_i$, of size $\frac{m}{2} \times \frac{n}{2}$ is generated for each frame of the video. For a generic i-th frame $\mathbf{f}_i$, of a video sequence, the final watermark $\tilde{\mathbf{w}}_i$ is obtained by multiplying $\breve{\mathbf{w}}_i$ with the pseudo-noise image, $\mathbf{p}_i$:

$$\tilde{\mathbf{w}}_i = \breve{\mathbf{w}}_i . * \mathbf{p}_i \tag{3.2}$$

where $.*$ represents element-by-element multiplication. Note that $\tilde{\mathbf{w}}_i \in \{-1, 1\}$.

The computed DCT coefficients of the luminance channel of the frame $\mathbf{f}_i$ are denoted as $\mathbf{F}_i$. The watermark, $\tilde{\mathbf{w}}_i$ is then scaled by a factor α, and added to a set of coefficients in $\mathbf{F}_i$ starting at the initial frequencies of (Δ_1, Δ_2). The resulting image $\mathbf{Y}_i$ is given by

$$Y_i(k + \Delta_1, l + \Delta_2) = F_i(k + \Delta_1, l + \Delta_2) + \alpha \cdot \tilde{w}_i(k, l) \tag{3.3}$$

where k and l correspond to the pixel location in the spatial domain and the coefficient location in the DCT domain. Here, $Y_i(\cdot, \cdot)$, $F_i(\cdot, \cdot)$, and $\tilde{w}_i(\cdot, \cdot)$ represent the individual

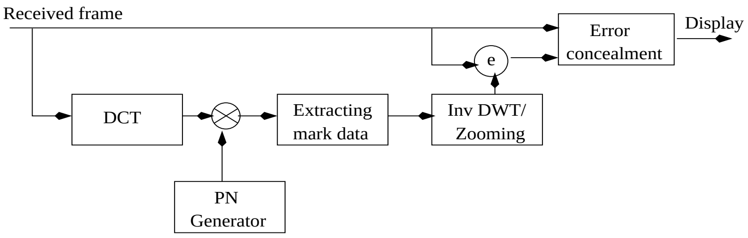


Figure 3.2: Block diagram of the retrieval algorithm.

component values of matrices $\mathbf{Y}_i$, $\mathbf{F}_i$, and $\tilde{\mathbf{w}}_i$, respectively. Note that $\Delta_1 \in [0, \frac{m}{2}]$ and $\Delta_2 \in [0, \frac{n}{2}]$. $\mathbf{Y}_i$ is then inverse transformed, encoded and transmitted.

In the proposed method, the final watermark is added only to the mid-frequency DCT coefficients. The range of frequencies where the watermark is inserted is strongly dependent on the application. For the purpose of delivering a high quality video through a channel and for better performance in error concealment, the mid-frequencies are a good choice. Inserting the watermark in the low-frequencies would cause visible artifacts in the image, while inserting it in the high frequencies would make it more prone to channel induced defects. Also, multiple copies of the marker can be inserted sequentially with various initial frequencies to make the watermark more robust to channel errors. In this case, the multiple copies are generated using independent randomly generated pseudo-noise matrices.

3.2.2 The Retrieval Part

The block diagram of the retrieval technique is shown in Fig. 3.2. The DCT coefficients of the luminance channel of the received frame $\mathbf{y}_{ri}$, denoted by $\mathbf{Y}_{ri}$, are computed as

$$\mathbf{Y}_{ri} = DCT_2(\mathbf{y}_{ri}) \tag{3.4}$$

where DCT_2 represents the 2-D DCT operation.

These coefficients are then multiplied by the corresponding pseudo-noise image $\mathbf{p}_i$. The pseudo-noise image generated is the same as that at the transmitter side. It is tacitly assumed that the receiver knows the seed for generating the pseudo-noise image and the initial frequencies, (Δ_1, Δ_2), where the mark was inserted. An issue of concern with this assumption is that it might lead to possible synchronization problems when severe channel errors cause loss of frames. This can in turn be handled by embedding the frame order number i (similar to the sequence number in packet transmission) into the frame itself. The receiver side pseudo noise generator algorithm can be driven by the recovered value while the missing frames can be detected using the missing frame order numbers.

The result of the multiplication, denoted as $\breve{\mathbf{w}}_{ri}$, is averaged over the 4 pixels (2×2 matrix form) and the binary marker is extracted by taking the sign of this average:

$$w_{ri}(k,l) = sgn\left\{\frac{1}{4}\left(\lambda_i(k,l)\right)\right\}. \tag{3.5}$$

where

$$\lambda_i(k,l) = \sum_{k'=2k-1}^{2k} \sum_{l'=2l-1}^{2l} Y_{ri}(k'+\Delta_1, l'+\Delta_2) \cdot p_i(k',l') \tag{3.6}$$

and $Y_{ri}(\cdot,\cdot)$, $p_i(\cdot,\cdot)$, and $w_{ri}(\cdot,\cdot)$ are the individual component values of matrices $\mathbf{Y}_{ri}$, $\mathbf{p}_i$, and the extracted marker $\mathbf{w}_{ri}$, respectively. Note here that the values of $\mathbf{w}_{ri}$ greater than 0 are assigned a value of 1 and those that are equal to or less than 0 are assigned a value 0 to make the resulting image binary. Also note that while the size of $\breve{\mathbf{w}}_{ri}$ is $\frac{m}{2} \times \frac{n}{2}$, the size of $\mathbf{w}_{ri}$ is $\frac{m}{4} \times \frac{n}{4}$.

It has been shown that this approach enables a fairly large amount of hidden data to be embedded without significantly affecting the perceptual quality of the encoded image [58]. Once the binary marker is extracted, the reduced size image is obtained by inverse half-toning the watermark.

Although a number of algorithms have been proposed for inverse half-toning [77]-[79], the inverse half-toning algorithm using wavelets proposed by Xiong *et al.* [77] is employed here because of its performance and ease of operation. This process primarily involves edge extraction from the high frequency components and edge preserving noise removal of the low frequency components of the wavelet coefficients. A discrete dyadic wavelet transform using 'Haar' wavelet (as a perfect reconstruction filter) without sampling rate conversion is employed to obtain back the processed smooth marker. The whole operation can be represented as

$$\hat{\mathbf{m}}_i = HT^{-1}(\mathbf{w}_{ri}) \tag{3.7}$$

where HT^{-1} denotes the inverse half-toning operation and $\hat{\mathbf{m}}_i$ is the extracted approximation of $\mathbf{m}_i$.

A 2-D inverse DWT is performed on this smooth marker to obtain an intermediate resolution image $\hat{\mathbf{g}}_i$. The values of $\hat{\mathbf{m}}_i$ form the approximation coefficients. Other high frequency coefficients are assumed to be 0 while computing the inverse DWT.

$$\hat{\mathbf{g}}_i = IDWT_2(\hat{\mathbf{m}}_i). \tag{3.8}$$

Note that $\hat{\mathbf{g}}_i$ is of size $\frac{m}{2} \times \frac{n}{2}$. It is then up-sampled by a factor of 2 and passed through a lowpass interpolation filter to obtain an $m \times n$ image. The resulting image $\hat{\mathbf{f}}_i$, is compared with the current received frame $\mathbf{y}_{ri}$ to detect and conceal the corrupted blocks by substituting the appropriate data.

The criterion for substituting the loss areas is different for images and video. The substituted areas in images are identified by packet-size blocks of lost data while in case of video, these are located using motion vectors and motion compensated error residual (MCER) values. An implementation issue here is that the marker needs to be scaled before the appropriate areas are substituted. The scaling can be either done throughout the image or only in the localized areas where the frame experienced packet losses. A global scaling constant is used when the image is globally scaled. In the case of local scaling, different local scaling factors are used based on the intensities of the surrounding areas. Both approaches are explored and the results presented in Section 3.4.

3.3 Implementation Scenarios

In this section, two different extensions of the algorithm with minor modifications are presented and discussed. The extensions to the regular implementation are: (1) Extension from wired to wireless transmission, and (2) Extension from gray scale to color images. The various models that are adopted to test the feasibility and effectiveness of the proposed algorithm in these two scenarios are also discussed.

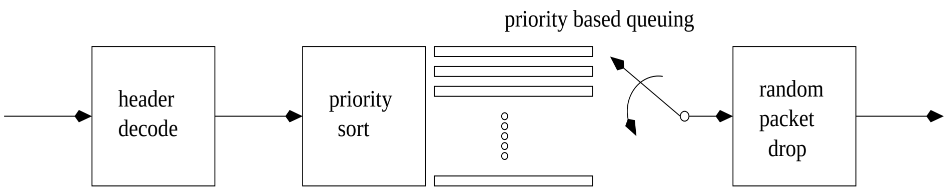


Figure 3.3: Block diagram of the lossy wired transmission model.

3.3.1 Wired vs. Wireless Transmission

The lossy wired model that has been adopted for implementation of the proposed algorithm is shown in Fig. 3.3. The testing of the algorithm can be easily performed at the IP level. The embedded image is packetized with appropriate protection and header information and transmitted.

At the router level, the following algorithm is implemented. The channel (layered) header is decoded to check for a match in the source, next hop, and destination IP addresses. Then the packets are sorted into one of the multiple priority queues based on the value of *priority_byte* in the header. The transmission from this point forward is based on the priority of the packet and the current channel conditions. Only when queue 1 is empty, packets from queue 2 are transmitted and so on. According to this model, packet loss is introduced when the queue buffer is full for each of the queues except for queue 1.

Once the packets are reordered based on their priorities, a delay is created in the transmission according to predefined latency values. The packets are then randomly dropped in accordance with a known probabilistic distribution (Gaussian in this case) which has a preset (controllable) mean and variance. The remainder of the packets are forwarded to the destination in a point-to-point network.

However, for wireless cases, the model in Fig. 3.3 does not work due to the following reasons: (1) Wireless channel has unpredictable variation with time, (2) Co- and cross-channel interferences are not accounted for, and (3) Fading and power losses are not considered in the current model. Apart from these, the model used in case of wired transmission is adapted for network traffic characteristics of Internet, typically like network congestion, which is quite unlike the case in wireless transmission.

A link layer modelling instead of network layer is adopted for wireless channel trans-

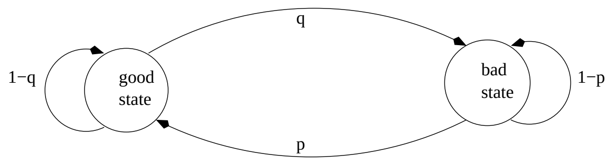


Figure 3.4: States of the adopted wireless simulation model.

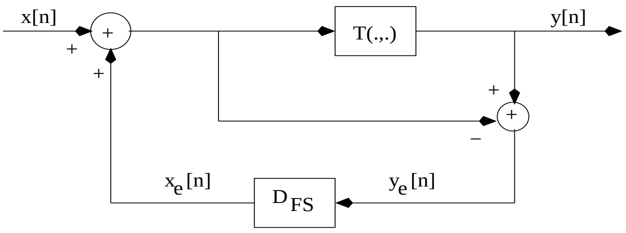


Figure 3.5: Block diagram of the Error Diffusion algorithm. D_{FS} is the Floyd-Steinberg kernel and $T(\cdot,\cdot)$ is the threshold operator.

mission scenarios. A simple point-to-point, two-state Markov chain, i.e., the Gilbert-Elliot model shown in Fig. 3.4, has been adopted for wireless transmission scenarios [80]. The two states in this model can be considered as the *good* state and *bad* state or the receive state and the loss state, respectively, with predefined probabilities p and q. This means that the Markov chain is in the *good* state if the packet is received in time without any errors and is in the *bad* state if the packet is lost during transmission due to latency or bandwidth limitations of the channel.

The parameters p and q are called the transition probabilities of the Markov chain between the *good* and the *bad* states. The transition matrix of this two-state Markov chain can be represented as

$$\mathbf{M}_2 = \begin{bmatrix} 1-q & q \\ p & 1-p \end{bmatrix} \tag{3.9}$$

The values of p and q are quite apart with more emphasis on the *good* state. For a typical wireless channel, the values of p and q would be around 0.999 and 0.001, respectively. Note here that $p + q = 1$. Such a model is followed here for the wireless channel simulations and the results are presented in Section 3.4.

3.3.2 Gray Scale vs. Color

The algorithm can be extended to work in the case of color image/video transmission. In this case, differences can be seen not only in the algorithm implementation, but in the results too. The variation in results and their analysis is presented in Section 3.4 while the implementation changes are discussed here.

The half-toning technique used for gray scale image processing is a simple feedback based loop shown in Fig. 3.5. Here, $\mathbf{D}_{FS}$ is the Floyd-Steinberg kernel given in Eq. (3.1) and $T(\cdot,\cdot)$ is a threshold operation. $x_e[n]$ and $y_e[n]$, the errors in the n-th sample of $x[n]$ and $y[n]$ respectively, are "diffused" back into $x[n+1]$ to $x[n+4]$ samples using a feedback loop. This implementation gives a pretty good estimate of the original gray scale image as a binary image.

For the case of color images/videos, we use color dithering [81]. Here, the color space is divided into 4 subspaces Cyan (C), Magenta (M), Yellow (Y), and black (K). The original RGB color image $\mathbf{S}_1$ is converted into an CMYK image $\mathbf{S}_{2new}$ using the transformation given by

$$\begin{aligned} \mathbf{S}_{2old} &= 1 - \mathbf{S}_1 \\ K &= min\left\{[\mathbf{S}_{2old}]^T\right\} \\ \mathbf{S}_{2new} &= \mathbf{S}_{2old} - K \end{aligned} \tag{3.10}$$

where $\mathbf{S}_1 = [R\ G\ B]^T$ and $\mathbf{S}_2 = [C\ M\ Y]^T$. The black values K are initially calculated using *old* C, M, Y values and then, the *new* C, M, Y values are obtained by modifying the *old* ones using these K values. The error diffusion shown in Fig. 3.6 is then applied to the CMYK images individually to obtain a color half-toned image.

The block diagram of the generalized error diffusion algorithm used for color image half-toning, shown in Fig. 3.6, is quite similar to Floyd-Steinberg dithering technique in Fig. 3.5 except that it has another positive feedback loop [82]. The sample $y[n]$, created from threshold operation on $x[n]$, goes through a scaled lowpass filter $\mathbf{A}$ with a predefined *hysteresis* value h. Since the sample values after threshold process are sharper in the CMYK space, this lowpass filtering operation is required. $\mathbf{A}$ is chosen to be a 3×3 lowpass filter to simplify implementation with respect to the 3×3 Floyd-Steinberg

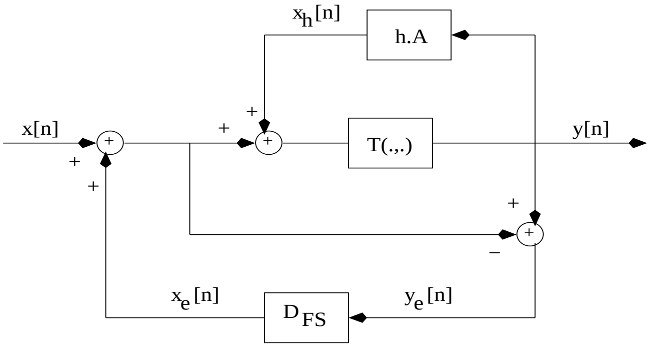


Figure 3.6: Block diagram of the Generalized Error Diffusion algorithm. h is the hysteresis value and $\mathbf{A}$ is a lowpass filter.

kernel $\mathbf{D}_{FS}$. h is chosen to be 0.5 in our implementation.

After the error-diffusion process, the image is converted back to RGB color space by an approximate inverse of the transformation defined in Eq. (3.10). The conversion can be represented as:

$$\begin{aligned} \mathbf{S}_{2new} &= \mathbf{S}_{2old} + K \\ \mathbf{S}_1 &= 1 - \mathbf{S}_{2new}. \end{aligned} \tag{3.11}$$

Either the R, G, and B channels or the Y, Cb, and Cr channels of the half-toned color marker signal can now be embedded into the luminance component of the original image/video frame. Each of these cases with individual and dependent variations in α are considered and the results are presented and analyzed in Section 3.4.

3.4 Experimental Results

The algorithm proposed in Section 3.2 with extensions proposed in Section 3.3 has implemented. For wired channel transmission, conventional UDP protocol with packet loss is considered, whereas a simulated point-to-point lossy link layer model is used for wireless cases. The following assumptions are made for simplicity with regard to the implementation of the algorithm: (1) The binary loss probability of the channel is assumed to be constant for a given network bandwidth, (2) The source transmission rate is assumed to be less than the maximum channel bandwidth, (3) No re-transmissions occur, and (4)

Bit errors over successfully received packets are negligible.

'Haar' wavelet is used for calculating the two-level approximate wavelet coefficients. For wired scenarios, the compressed output stream is vectorized and multiplied with a vector that is randomly generated using a Gaussian distribution. About 1000 varying packet loss simulations have been generated independently for each transmission and their statistical average is taken to obtain the probability loss vector.

In the case of wireless transmission, the channel and its losses are simulated using the *ns-2* simulator [83]. The power levels for the wireless transmission are kept almost constant. A constant fading is assumed for each of the packet transmissions in the individual channels of a multichannel scenario. It should be noted here that a standard CDMA2000 system [48] is followed due to its efficiency of operation in the link layer when compared to other wireless systems.

Table 3.1 summarizes the results of the experiment for wired and wireless transmissions. In the case of color images, the Composite PSNR (CPSNR) is calculated at the receiver. The symbol * indicates the images/video frames that are implemented using the wired transmission simulations. Note that the percentage improvement due to localized error concealment in the case of wireless transmission is much more pronounced than in the case of wired transmission. For each value of image/video frame, the PSNR of the received image (PSNR_{rec}), the PSNR of the error concealed image (PSNR_{erc}), and the PSNR of the local-scaled error concealed image (PSNR_{loc}) are noted.

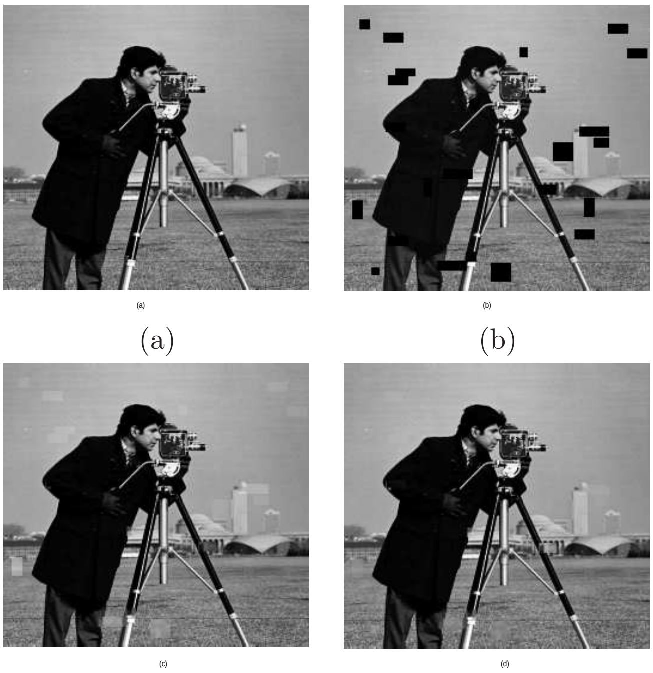
When the received frame is error concealed using the proposed algorithm, the scaling operation performed on the concealed data blocks is fixed. Instead, if localized interpolation is performed on the reconstructed blocks by considering the neighboring pixels that were correctly received, an improvement in the visual quality of the restored frame is observed (as can be seen in Table 3.1).

A sample result for the wired transmission case is shown in Fig. 3.7 for the *Cameraman* image with the parameter values: $\alpha = 3.6$, mean loss probability = 0.15, loss variance = 2.5%. The received frame had a $\text{PSNR}_{rec} = 19.83$ and the error concealed image had a $\text{PSNR}_{erc} = 27.69$. These frames with the original frame are shown in Figs. 3.7(a), (b) and (c). Fig. 3.7(d) has been obtained by localized scaling error concealment. Here, the error-concealed image is locally scaled by using a localization kernel of size 8×8 or

Table 3.1: Performance of the proposed algorithm. PSNR (in dB) for a fixed mean loss 15% and variance 2.5%

Frame/Image	PSNR$_{rec}$	PSNR$_{erc}$	PSNR$_{loc}$	% inc
Sail	19.4923	26.5968	29.6880	11.6
Cameraman*	19.8271	27.6852	28.9047	4.4
News	21.9273	27.2865	30.5244	11.9
Psycho	15.6776	25.4005	29.4129	15.8
Table tennis	16.8707	24.3020	27.6268	13.7
Flower	17.8046	21.6195	24.6611	14.1
Football	19.1367	27.7806	30.9214	11.3
Stefan	20.1159	26.5788	29.4279	10.7
Coastguard*	20.0866	25.6650	27.0271	5.3
Hockey	19.3017	24.3303	33.2081	36.5
Girl	18.1319	25.4701	31.7092	24.5
Frank	20.0172	23.0964	29.0501	25.8
Surf	18.1977	23.5479	30.4909	29.5
Gold hill*	18.0111	27.1874	28.4486	4.6

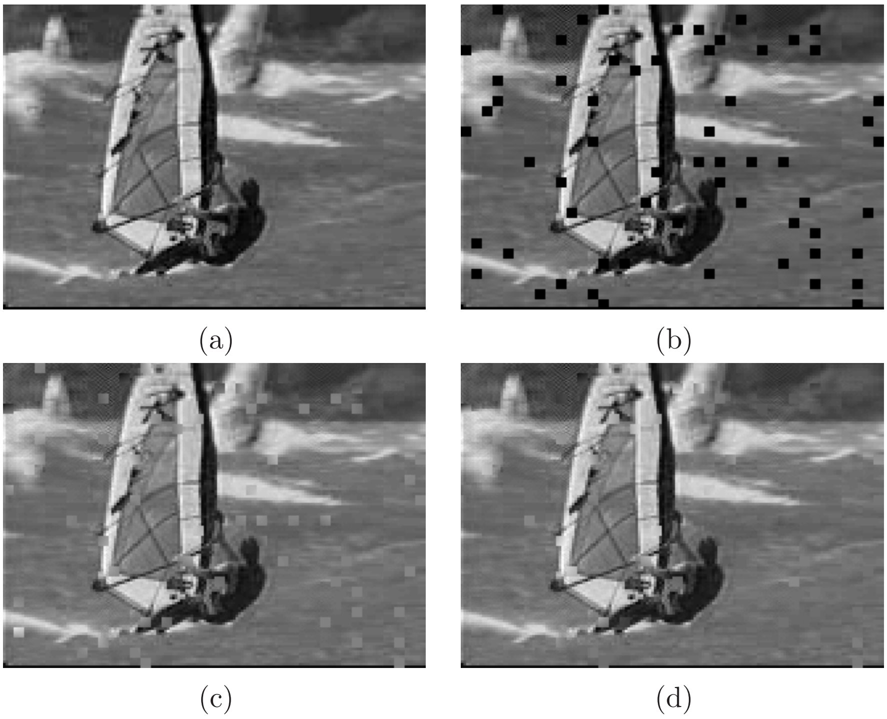
*wired cases



(a) (b)

(c) (d)

Figure 3.7: (a) original frame, (b) received frame with mean loss probability = 0.15, variance = 2.5%; PSNR = 19.8271 dB, (c) error concealed frame; α = 3.6; PSNR = 27.6852 dB, and (d) localized scaling error concealed frame; PSNR = 28.9047 dB.



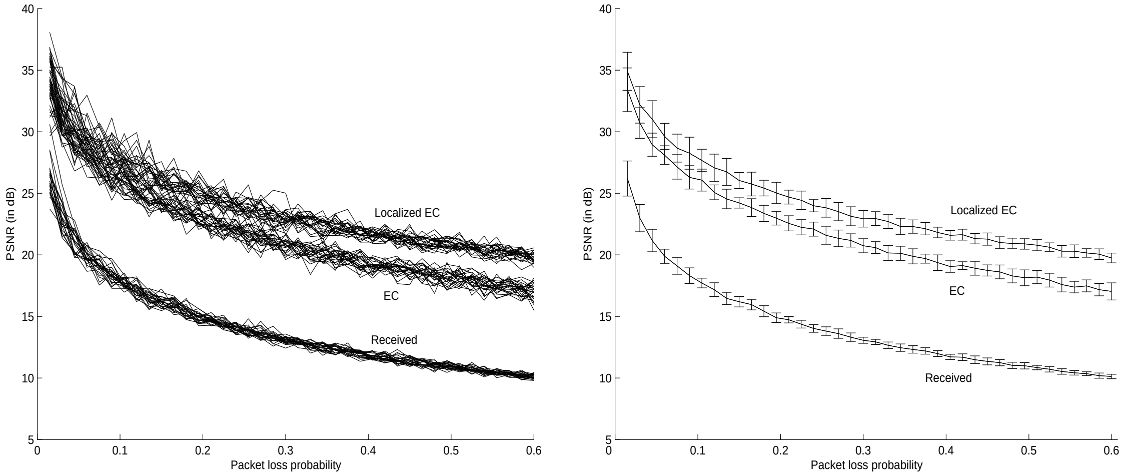
(a) (b)

(c) (d)

Figure 3.8: (a) original image, (b) received image with mean loss probability of 0.12 and variance 5% (PSNR = 19.3003 dB), and (c) error corrected image using the data hiding algorithm with $\alpha = 5$; (PSNR = 28.1702 dB), and (d) The localized error corrected image with kernel size 8×8; (PSNR = 29.9137 dB).

16×16 (based on the size of the lost data area). The PSNR obtained by performing this localized scaling operation on the error-concealed image is $\text{PSNR}_{loc} = 28.90$. However, better results in terms of PSNR are expected if the kernel size is varied.

In Fig. 3.8, a sample result for the wireless transmission simulation are presented for the *Sail* image with the parameter values: $\alpha = 5$, average loss probability = 0.12, and loss variance = 5%. The received image had a $\text{PSNR}_{rec} = 19.30$ (the value of 19.4923 that is listed in the table is CPSNR, i.e., it is obtained for the color image). The original, received and error-concealed images ($\text{PSNR}_{erc} = 28.17$) are shown in Figs. 3.8(a), (b) and (c) respectively and the localized error concealed image with 8×8 kernel ($\text{PSNR}_{loc} = 29.91$) is represented in Fig. 3.8(d).



(a)

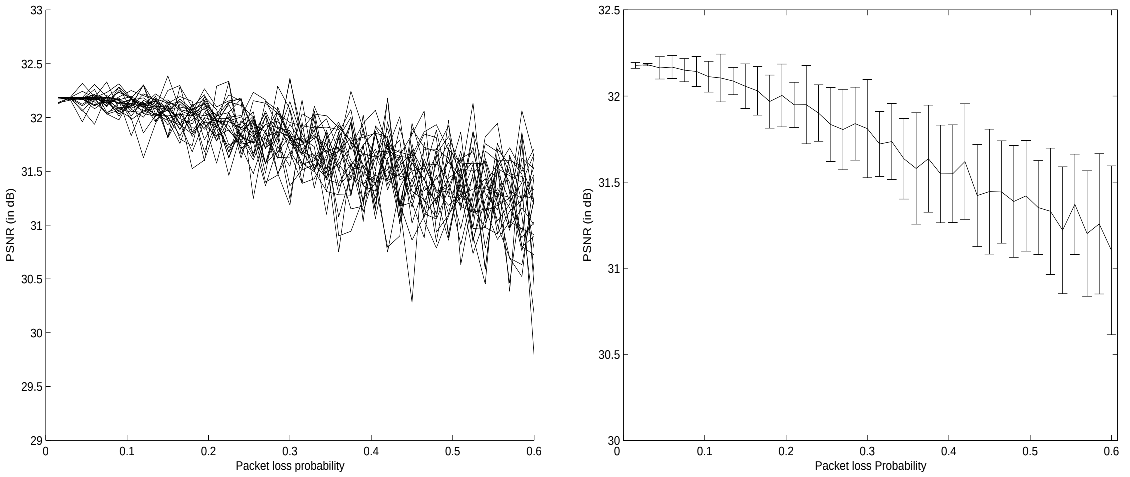
(b)

Figure 3.9: (a) Comparison of the quality of the received frame vs. the error concealed (EC) and the localized error concealed frames with variation in loss probability for a sample set of 25 simulations and (b) mean and standard deviation of the curves in (a).

3.5 Analysis

A sample set of 25 iterations for error concealment and localized concealment algorithms on *Sail* image (due to randomness in loss for wireless scenarios) is represented in Fig. 3.9(a) with variations in packet loss probability. As seen from these simulations, the quality of the error-concealed image is much better (approximately $6 - 8$ dB improvement) for all cases. The localized error concealment algorithm achieved higher quality (approximately $2 - 3$ dB over the error concealment algorithm). Note that the improvements in the PSNR values increased with increasing loss probabilities. The mean and standard deviations of these curves are shown in Fig. 3.9(b). Note that the variation of standard deviations was higher at lower packet losses and lower at higher packet losses.

An interesting aspect to observe from this graph is that the standard deviation of the received signal is large when the packet losses are very small and reduces as the packet loss percentages increase. This is rather tricky to understand. Consider an image with a lot of low and high frequency detail. When the packet loss percentages are small, the losses could occur in the smooth (low frequency) areas or in the high frequency areas. If the loss occurs in the smooth areas, the PSNR value changes would be low considering the neighboring blocks around the lossy area. If however, the packet losses occurred in

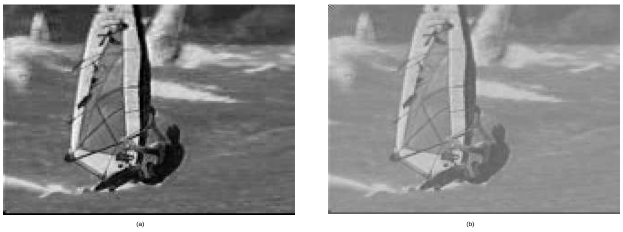


(a) (b)

Figure 3.10: (a) The variation of the quality of the extracted watermark with increasing packet loss probabilities for the same sample set as in Fig. 3.9(a). (b) The mean and standard deviation of the extracted watermark quality in (a).

the high frequency area, there would be a substantial decrease in the PSNR values. This change increases the standard deviation when the losses are low. On the other hand, when losses are high, both the low and the high frequency areas will be equally damaged thereby maintaining low variation in the decrease of PSNR values. Therefore, over an ensemble set of 25 values, this standard deviation of the received image PSNR decreases as the packet losses increase.

Since four copies of the marker are embedded in the image/video frame, it can be extracted with high quality even at higher loss probabilities of $0.4 - 0.6$. The variation of the watermark quality with the loss probability for the same sample set of 25 simulations is plotted in Fig. 3.10(a). The mean and the standard deviations of the extracted watermark quality are plotted in Fig. 3.10(b). Note that while the mean decreased, the standard deviations increased non-linearly with increasing packet loss probabilities. This suggests that the variation in the decrease in extracted watermark quality is non-linear, i.e., the decrease in quality varies less at lower loss probabilities and changes more at higher loss probabilities even with high robustness. It can be shown that finding an optimum value of the scaling parameter, α, might compensate for this non-linear variation.



(a) (b)

Figure 3.11: Results with two different values of α: (a) $\alpha = 1$, and (b) $\alpha = 10$ (an extreme case).

3.5.1 Scaling Parameter Variation

The value of the scaling parameter α is changed to vary the strength of the embedded watermark. The parameter α in this case defines the robustness of the watermark with respect to channel errors. As α increases, it not only enhances the quality of the received marker, but also makes the embedded watermark more visible thus reducing the quality of the received image/frame. Hence, it is important to choose an optimum value of α such that there is a balance between the quality of the received image/video frame and the amount of the information embedded for the quality of the received watermark to be high.

The results of the experiment with two different values of α can be seen in Fig. 3.11. The *Sail* image in Fig. 3.11(a) is embedded with α value of 1 while that in Fig. 3.11(b) is embedded with α value of 10 as an extreme case. The embedded signal content in Fig. 3.11(b) increased multifold thus highly decreasing the quality of the frame.

The quality of the extracted watermark from the received image, however, has quality variations (given by the PSNR values) that follow a different trend. Fig. 3.12 shows the variation of the quality of the extracted watermark for packet loss probability = 0.09. Note that the quality (PSNR) converges to a value of 32.11 dB in the case of *Sail* image. The value of α at which this maximum is achieved, α_{thresh}, can be defined as the smallest α after which there is no substantial increase in the quality for increasing α. The value of α_{thresh} for the *Sail* image is 4.74.

Since increasing α would decrease the quality of the received image/video frame, choosing $\alpha = \alpha_{thresh}$ would be a start in finding the best value of α that would maximize the quality of the error concealed image. However, even at α_{thresh}, we are not guaranteed

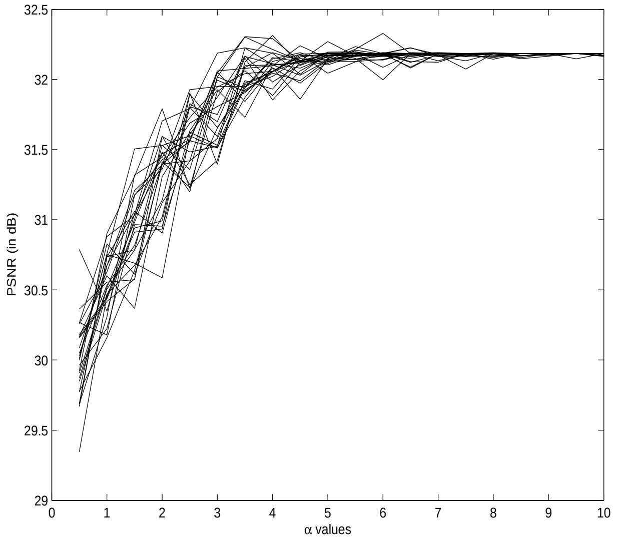


Figure 3.12: Result of watermark quality variation with α for a sample packet loss probability of 0.09. The plot also shows the convergence of watermark quality for higher values of α.

a high quality of the received image. Here, the best value of α is found graphically by plotting the quality variations of the received image and the extracted watermark as a function of α and finding the point of intersection of these two curves based on the assumption that the transmitter has a good estimate of the channel losses (this can be achieved by either using the feedback from the receiver or expecting the channel to follow a particular probabilistic distribution that closely approximates the real time channel losses).

Fig. 3.13 shows this plot for the *Sail* image for a mean packet loss probability of 0.09. As expected, the best value of α (the point of intersection, 4.19) lies below the value of α_{thresh}. Note that this value is not the best with regard to the host image quality or the detected watermark quality. The value of α_{thresh} gives the best overall error concealment performance. For obtaining the highest host image quality by considering the trade-off with regard to the detected watermark quality, the "knee" of the curves in Fig. 3.13. Also, since this value is close to α_{thresh}, the non-linear marker quality fluctuations seen in Fig. 3.10(b) are reduced if not eliminated.

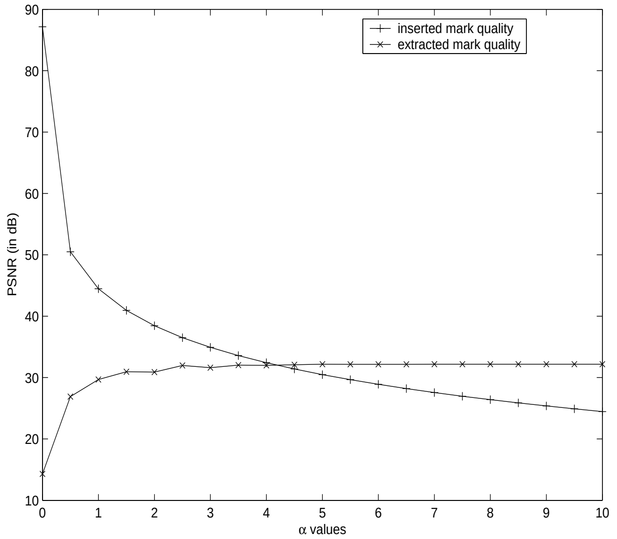


Figure 3.13: Comparison of extracted watermark quality with degrading frame quality after watermark insertion.

3.5.2 Effects on Chrominance Components

For the case of color images/video, losses are simulated both in the RGB color space and the YCbCr color space. In the latter case, even though the losses affecting all three channels are considered, those that affect only the luminance channel are given higher priority because most of the embedded information in the YCbCr (YUV in case of H.263) color space is mainly concentrated in the luminance channel.

The result of the algorithm implementation on the *Sail* color image are shown in Fig. 3.14. The original color image is shown in Fig. 3.14(a), while Fig. 3.14(b) represents the received image with errors due to wireless channel occurring in all three channels of YCbCr color space ($\text{PSNR}_{rec} = 19.49$). The error concealed image with $\text{PSNR}_{erc} = 26.60$ and the localized scaled error concealed image with $\text{PSNR}_{loc} = 29.69$ are shown in Figs. 3.14(c) and (d), respectively.

Here, the embedded color dithered watermark is in RGB color space and all three channels are embedded into the luminance channel of the image. The watermark can also be converted into YCbCr color space before it is embedded. This way, the implementation of the algorithm at the receiver is much simpler. Besides, it helps in the compression of data to be embedded. For example, instead of embedding the entire color information, we can sub-sample the chrominance components and embed the 4:2:2 or 4:2:0 watermark

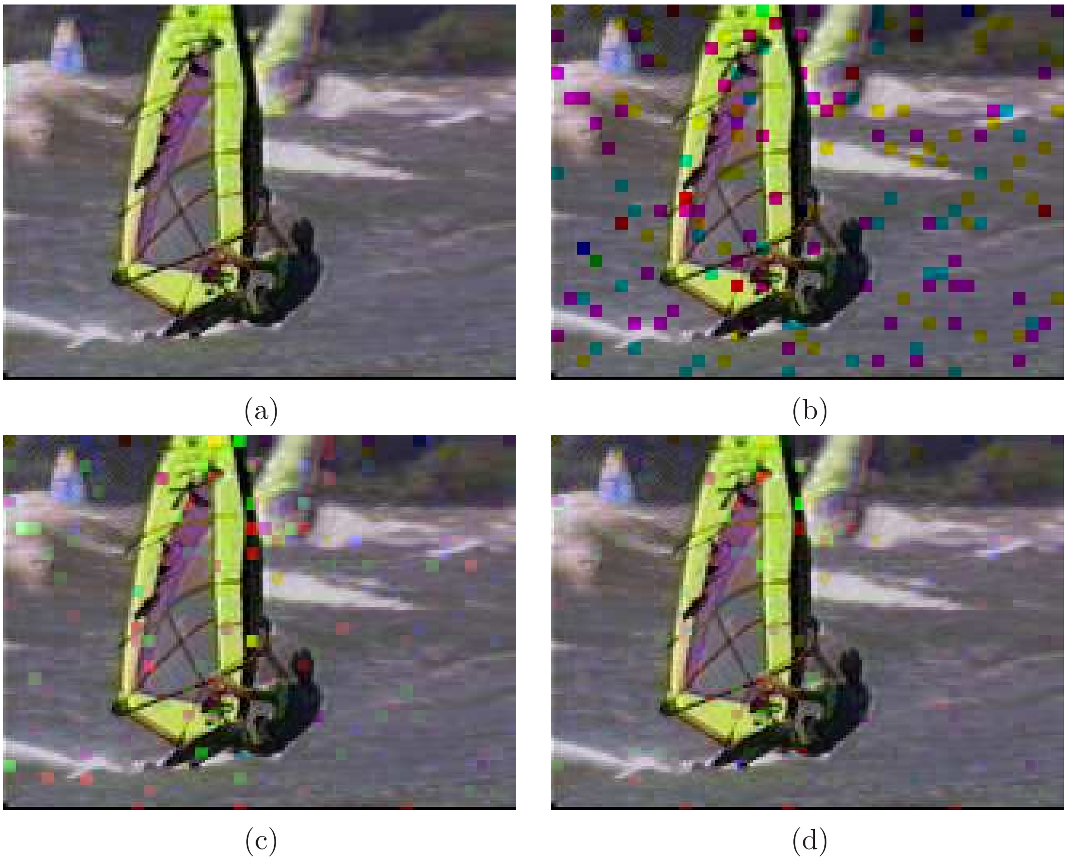
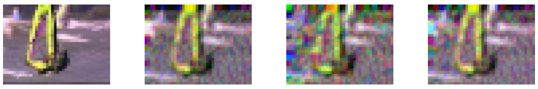


Figure 3.14: (a) original color image, (b) received image with mean loss probability of 0.15 and variance 3% with loss occurring in all 3 channels (PSNR = 19.4923 dB), (c) error concealed image using the data hiding algorithm (PSNR = 26.5968 dB), and (d) localized error corrected image (PSNR = 29.6880 dB).

in the luminance channel of the original image/video frame. This proves to be effective for codecs like H.263 where such kind of sub-sampling is usually adopted.

The extracted watermark quality difference between embedding the RGB channels of the watermark and embedding the YCbCr channels is very little. This can be seen from Fig. 3.15. The original color watermark, the extracted color watermark from the received image (mean loss probability = 0.15, loss variance = 5%) when RGB channels of the watermark are embedded (α = 5), and when YCbCr channels are embedded (α = 5) are shown in Figs. 3.15(a), (b) and (c), respectively. In fact, the simplicity in the implementation and compression are the only reasons that make embedding sub-sampled chrominance components more appealing.

The quality of the extracted color watermark can be further improved if different α val-



(a) (b) (c) (d)

Figure 3.15: (a) original watermark color image, (b) received watermark for a mean loss probability of 0.15 and variance 3% when RGB channels are embedded with $\alpha = 5$, (c) extracted watermark for the same loss when YCrCb channels (4:2:0) are embedded with $\alpha = 5$, and (d) extracted watermark when YCrCb (4:2:0) channels are embedded with $\alpha_Y = 3.75$, $\alpha_{Cb} = 4.5$, and $\alpha_{Cr} = 5$. Note that (b), (c), and (d) are inverse half-toned images.

Table 3.2: Performance comparison with PSNR (in dB).

Image	Sun's	Salama's	Wang's	Park's	Li's	Ours
Lena	23.93	23.99	24.41	24.96	26.46	28.23
Pepper	22.19	23.69	24.06	24.48	27.25	29.47
Zelda	26.35	27.13	26.40	27.36	28.33	29.08
Baboon	17.46	18.98	19.02	17.42	-	21.92

ues are used to embed each of the Y, Cb, Cr channels independently. Fig. 3.15(d) shows the extracted color watermark when the α values for luminance and the chrominance components are 3.75, 4.5, and 5, respectively. In this case, a reasonable improvement is seen in the quality of the extracted watermark when compared to Figs. 3.15(b) and (c). The values of α_Y, α_{Cb}, and α_{Cr} are randomly chosen here and may not be optimum for each of the individual channel's reconstruction. However, better results are expected if such kind of optimization for the color channel scale factor is performed.

3.5.3 Comparison with Other Techniques

Table. 3.2 shows the PSNR comparison of the proposed technique and the techniques presented in [68]-[72] (obtained from [72]). Note that the proposed algorithm gives an improvement of 1-3 dB over the existing error concealment techniques.

An alternate method to achieve high level of error concealment performance is to transmit the low resolution version of the image/video frame as a encoded side information [84]. The encoded bits of the low resolution image would be generated in the same manner as proposed. They would comprise the side information and transmitted through the same channel instead of embedding them in the source. This approach yield results that are comparable to the proposed technique. However, there are three problems with this approach:

- Side information can be appended to the encoded video, but this addition incurs higher transmission bandwidth. If however, the compression or encryption of the video is increased to accommodate for the side information without increasing the bandwidth, there would be an increase in computational complexity. This can be clearly seen in Fig. 3.16. Data hiding provides a seamless alternative to achieve this trade-off [85].

- Since the effects of the wireless channel are approximated by independent or burst errors from packet losses, significant loss would occur in the low resolution reference. Therefore, recovering the lost data of the high resolution image would be very difficult if either the areas where the losses occur match or higher losses occur in the zoomed up low resolution image.

- More often than not, the compressed mark embedded image requires more bits to be encoded than the compressed original due to the higher entropy of the data hiding process. However, the increase in the number of bits is very small. For example, a *Hockey* video frame of size 240×320 was 7899 bytes after compression, while its marker was 1085 bytes. The mark embedded image was 8015 bytes after compression. This implies that the transmission of side information bits would require 969 bytes more (and therefore higher bandwidth) than the encoded data hidden signal.

Simulations have been carried out for the case of transmitting the low resolution image as a data hidden watermark (WEC) and as encoded side channel information (ECSI) in wireless scenario. Fig. 3.17(a) gives the PSNR versus loss rate curves of the two techniques. It can be seen that WEC performs better at higher loss rates (up to 3-4 dB improvement), while the performance of both techniques is comparable at lower loss rates (up to about 0.2). Note that at very low packet loss probabilities (< 0.04), ECSI tends to give a better performance. This may be due to the fact that the end result of

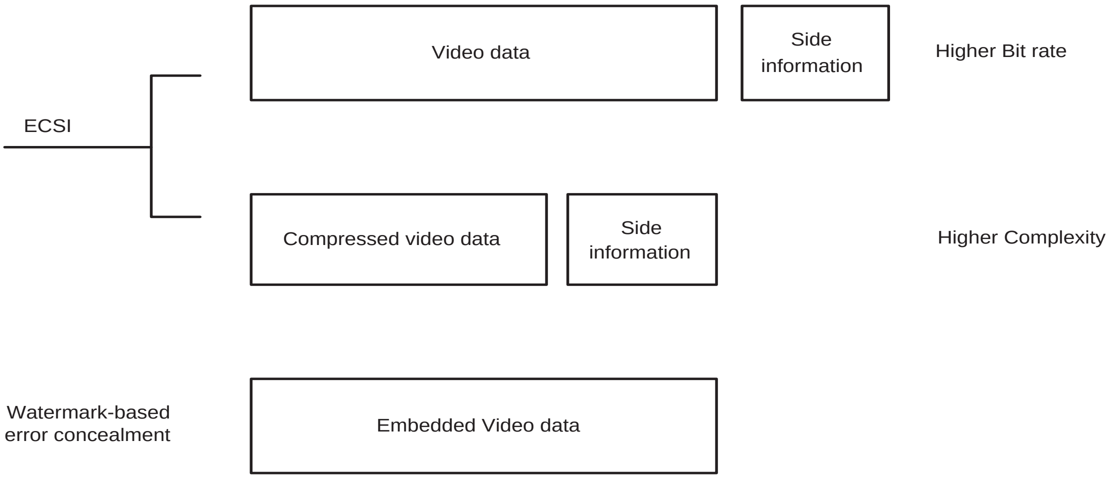


Figure 3.16: Illustration of comparison between WEC and ECSI.

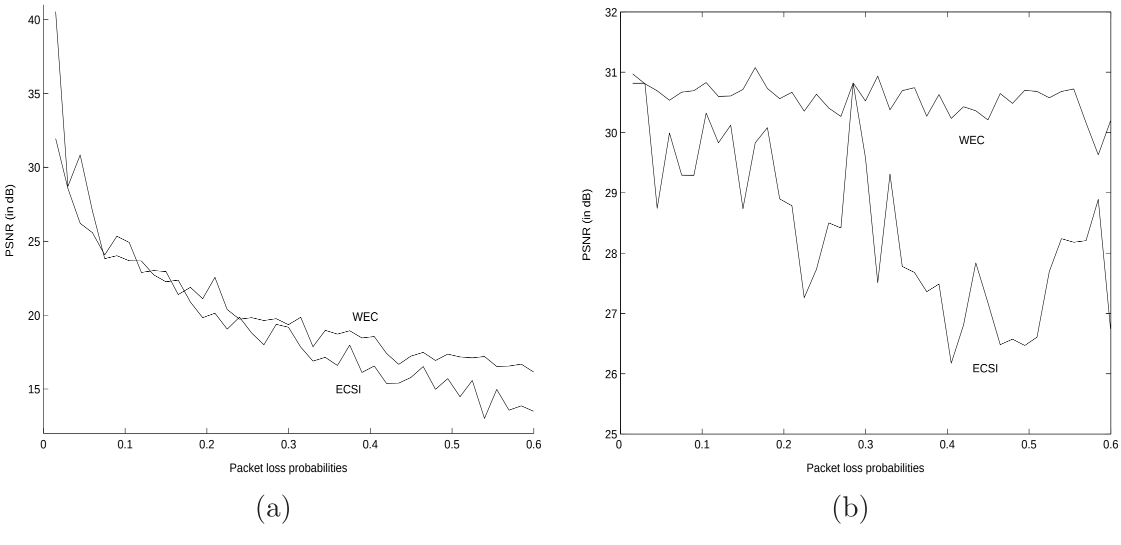


Figure 3.17: PSNR vs. loss rate curves for the techniques of WEC and error concealment using side information (ECSI) for the *Sail* image. (a) PSNR of the reconstructed image and (b) PSNR of the retrieved low resolution images.

the WEC suffers from minor data hiding defects, which are rarely visible. Fig. 3.17(b) shows the variation in the PSNR values of the received low resolution reference images.

3.6 DPCM Bit Stream Embedding

We next look at a variation of the proposed WEC algorithm wherein the binary watermark is generated in a different manner from the previously considered halftoned, low resolution version (obtained from the approximation coefficients from the 2D-DWT) of the host image. In this WEC scheme, we consider the embedded watermarked reference as the encoded energy content of the frame itself. A set of four 2D DCT coefficients of each block are obtained in a 2 × 2 matrix format and encoded using DPCM. A spread spectrum watermarking algorithm is used to embed the DPCM bit stream in the mid-frequency range of the video frame. To reduce the encoder complexity and cater to the feasibility issues, embedding operation is performed only in the intra-coded frames of the video. The algorithm compares closely with the WEC algorithm that embed the halftoned bits of the low resolution version of the video frame that we proposed earlier in this chapter [86]. We now provide a detailed comparison between these two techniques along with ECSI.

3.7 WEC for DPCM Encoded Child Image

The proposed scheme is divided into an embedding part and a retrieval part and each part is explained separately.

3.7.1 The Embedding Part

The data hiding technique used here is again a modified version of Cox's watermarking algorithm [73]. The block diagram of the embedding algorithm is shown in Fig. 3.18. In this technique, a block 2D DCT of a video frame is obtained and each block is quantized using the JPEG quantizer. A set of four coefficients for each block (one DC and three adjacent AC coefficients) is selected in a 2 × 2 matrix format. These coefficients are DPCM encoded and the encoded bit stream is ordered into a binary block, which forms

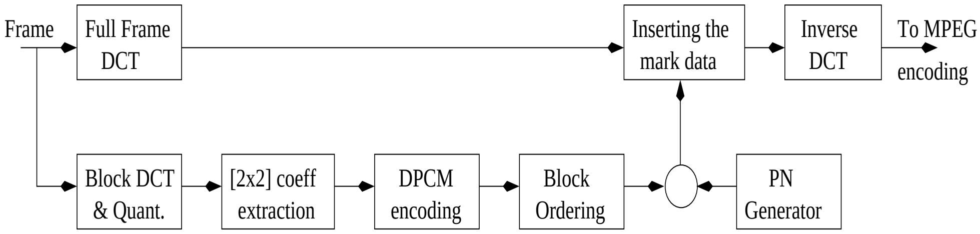


Figure 3.18: Block diagram of the embedding algorithm.

the 2D marker, $\mathbf{m}_i$, for the i-th key frame. One marker is used for each intra-coded frame. An identifier is added at the end of DPCM code of each block for synchronization and effective decoder operation.

The set of coefficients selected for embedding is based on the encoder embedding capacity, compression quality factor, DPCM code length, and size of the marker required. In our case, by using a 2×2 set of coefficients, the marker is of size $\frac{m}{2} \times \frac{n}{2}$ for a video frame of size $m \times n$, i.e., the marker is $\frac{1}{4}$-th the size of the video frame.

A unique zero mean unit variance random pseudo-noise image, $\mathbf{p}_i$, of size $\frac{m}{2} \times \frac{n}{2}$ is generated with a known seed for each intra-coded frame of the video. For a generic i-th key frame $\mathbf{f}_i$, of a video sequence, the embedding watermark $\mathbf{w}_i$ is obtained by multiplying $\mathbf{m}_i$ with the pseudo-noise image $\mathbf{p}_i$.

The computed DCT coefficients of the luminance channel of the frame $\mathbf{f}_i$ are denoted as $\mathbf{F}_i$. The watermark $\mathbf{w}_i$ is then scaled by a factor α, and added to a set of these coefficients starting at the initial frequencies of (Δ_1, Δ_2). The resulting image $\mathbf{Y}_i$ is given by

$$Y_i(K + \Delta_1, L + \Delta_2) = F_i(K + \Delta_1, L + \Delta_2) + \alpha \cdot w_i(k, l) \tag{3.12}$$

where k and l correspond to the pixel location in the spatial domain, and K and L correspond to the coefficient location in the DCT domain. Here, $Y_i(\cdot, \cdot)$, $F_i(\cdot, \cdot)$, and $w_i(\cdot, \cdot)$ represent the individual component values of matrices $\mathbf{Y}_i$, $\mathbf{F}_i$, and $\mathbf{w}_i$, respectively. Note that $\Delta_1 \in [0, \frac{m}{2}]$ and $\Delta_2 \in [0, \frac{n}{2}]$. $\mathbf{Y}_i$ is then inverse transformed, encoded and transmitted.

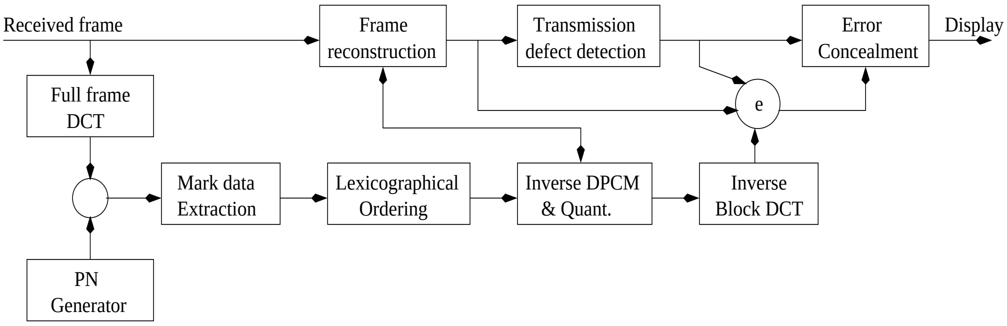


Figure 3.19: Block diagram of the retrieval algorithm.

3.7.2 The Retrieval Part

The block diagram of the retrieval technique is shown in Fig. 3.19. The DCT coefficients of the luminance channel of the received frame $\mathbf{y}_{ri}$, denoted by $\mathbf{Y}_{ri}$, are computed as

$$\mathbf{Y}_{ri} = \mathrm{DCT}_2(\mathbf{y}_{ri}) \tag{3.13}$$

where DCT_2 represents the 2-D DCT operation.

These coefficients are then multiplied by the corresponding pseudo-noise image $\mathbf{p}_i$. It is assumed that the receiver knows the seed for generating the pseudo-noise image and the initial frequencies, (Δ_1, Δ_2), where the mark was inserted. The binary marker is extracted by taking the sign of the result of the multiplication. This is given as

$$m_{ri}(k,l) = sgn\left\{Y_{ri}(K+\Delta_1, L+\Delta_2) \cdot p_i(k,l)\right\}. \tag{3.14}$$

where $Y_{ri}(\cdot,\cdot)$, $p_i(\cdot,\cdot)$, and $w_{ri}(\cdot,\cdot)$ are the individual component values of matrices $\mathbf{Y}_{ri}$, $\mathbf{p}_i$, and the extracted marker $\mathbf{m}_{ri}$, respectively. Note here that the values of $\mathbf{m}_{ri}$ greater than 0 are assigned a value of 1 and those that are equal to or less than 0 are assigned a value 0 to make the resulting image binary. Also note that the size of $\mathbf{m}_{ri}$ is $\frac{m}{2} \times \frac{n}{2}$. Furthermore, the right hand side of Eq. (3.14) is not averaged over the 4 pixel values as we are not embedding 4 copies of the image.

The extracted data is lexicographically ordered into a single bit stream. Here, it is assumed that the receiver is aware of the identifier code at the end of DPCM code of each block. This assumption can be validated by using the eob (end of block) identifier in JPEG-like encoding. It can also be encoded, embedded, and transmitted along with the child image. Since the identifier bits are embedded (in full frame DCT of the parent image) before compression, this would not de-synchronize the existing eob of compression code. The reference image is reconstructed by obtaining the inverse DPCM values and using them as a 2×2 set of coefficients of each block. Other AC coefficients are assumed 0 for each block. These block coefficients are inverse quantized and inverse block DCT transformed to obtain the reference $\hat{\mathbf{f}}_i$, which is used for error concealing the lossy parent image.

3.7.3 Comparison of ECSI, WEC with DPCM, and WEC with Halftoning

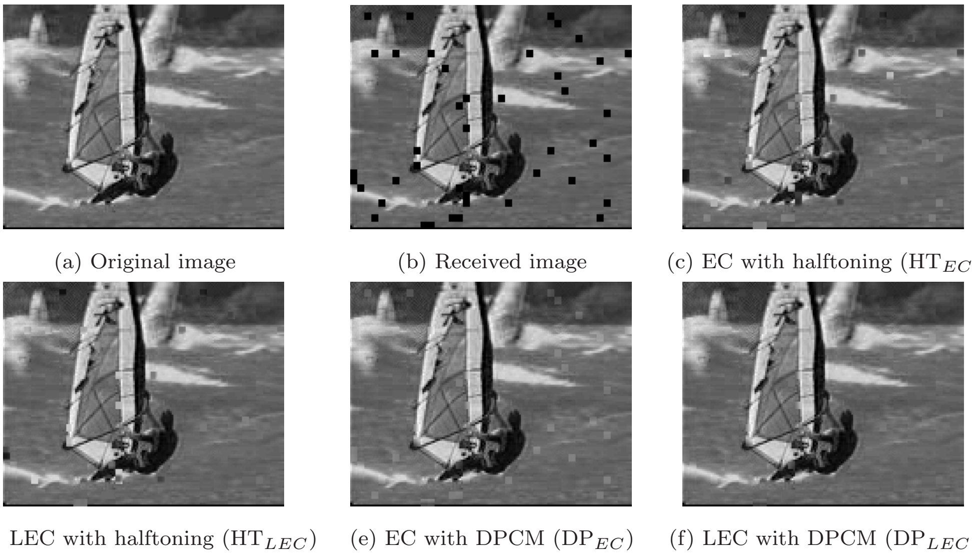
For a fixed packet loss probability, the bits required to compress the original video frame are fewer than the bits required to compress the embedded frame. However, this difference is small and therefore will not compensate for the bits required to compress or encode the side information. This means that the transmission of halftone or DPCM bits as side information would require more bits than the difference in compression of mark-embedded and unmarked frames. Also, any additional computational complexity advantage that ECSI has over WEC (since ECSI does not have to perform the embedding operation) is nullified by the complexity of encoding the side information. This can be seen in Fig. 3.16. Furthermore, since the side information bits are not as protected as the embedded bits, transmission loss incurs more errors in the received reference image in case of ECSI than WEC. If higher protection is given to the side information bits such that the performance of ECSI is comparable to that of WEC at higher packet loss probabilities, then the encoding complexity of ECSI increases. Therefore, WEC provides more optimal point in the performance-complexity curve than ECSI. An extended analysis of these trade-offs along with PSNR vs. loss rate curves is given in [86].

The advantages of using WEC with DPCM approach over WEC with halftoning are as follows: Firstly, fewer number of bits are required to be embedded. As a comparison, the number of bits required to embed the child image after halftoning are of the order of $36,000$ for an image of size 240×320. However, this is reduced to around $4,000$ in the case of DPCM bit stream embedding. In fact, this reduction allows us to embed the DPCM bit stream of the parent image with its full resolution instead of using a low resolution version as a reference. Secondly, the complexity of the transceivers are greatly reduced. Not only are the DWT and halftone operations not required at the encoder, but also the zooming and/or the inverse DWT operations in the receiver are not used. Instead, JPEG-like operations such as quantization and DPCM encoding of DPCM coefficients are used thus reducing the complexity of the codec. Thirdly, the performance of the error concealment and the localized error concealment algorithms are much improved over WEC with halftoning. This is because of two reasons: (1) the reference can be extracted to higher quality than the halftoned version, and (2) since fewer bits are used, α could be reduced and therefore the parent image would suffer from lesser watermarking artifacts. And lastly, since the DPCM bit stream of the reference image is embedded, it is more secure to outside attacks and transmission errors than just embedding halftone bits.

3.8 Simulations with DPCM Bit Stream

The experiments were performed with a sample set of videos of a fixed size of 240 × 320. The *ns-2* simulator was used to generate the wireless transmission loss scenarios. A Gilbert-Elliot loss model is used for generating the two-state Gaussian packet loss distribution with a predefined mean and variance. To simplify the experiment, it is assumed that: (1) single and 2-bit errors of successfully received packets were negligible, and if the number of bit errors exceed 2, the packet is lost, (2) no re-transmissions occur, (3) the loss probability is constant for a given channel bandwidth, and (4) packet size is fixed at 1 macro block (MB) for a given video transmission.

The embedding operation is performed with α value of 3. At the receiver, the packet loss areas are located and error concealed. In the discussion that follows, the frames that are error concealed in this manner are referred to as EC frames. Since only a set of coefficients are embedded, the reconstructed reference would not have high frequency contents. Therefore, the lossy areas in the video frame which have been error concealed would look smoother than the neighboring areas. For this reason, the high frequency



(a) Original image (b) Received image (c) EC with halftoning (HT$_{EC}$)

(d) LEC with halftoning (HT$_{LEC}$) (e) EC with DPCM (DP$_{EC}$) (f) LEC with DPCM (DP$_{LEC}$)

Figure 3.20: The original image is shown in (a). The received image is obtained for a mean packet loss probability of 0.15 and variance 2.5%. The PSNR values of the images are: (b) 20.2943, (c) 27.3112, (d) 30.8620, (e) 28.8613, and (f) 33.1337. The value of α used in both cases was 3.

Table 3.3: Algorithm performance in terms of PSNR (dB) for WEC with halftoning, WEC with DPCM and ECSI.

Video	HT$_{EC}$	HT$_{LEC}$	DP$_{EC}$	DP$_{LEC}$	ECSI
Foreman	25.83	29.52	27.31	30.77	29.13
News	26.42	29.89	27.95	31.06	28.68
Football	27.04	29.58	28.11	30.92	29.01
Flower	25.87	28.98	28.03	30.16	29.22
Hockey	25.35	29.44	27.39	31.27	29.08

content around a 16×16 pixel-area surrounding the loss are locally scaled to improve the EC image's perceptual quality. The frames that are processed in this fashion are referred to as LEC (local-scaled error concealed) frames.

Fig. 3.20 shows the performance of the proposed algorithm and that of WEC with halftoning. Figs. 3.20(a) and (b) show the original video frame and packet loss effected received frame respectively. Figs. 3.20(c) and (d) show the error concealed and localized error concealed frames obtained using WEC with halftone bit stream embedding, and Figs. 3.20(e) and (f) show the error concealed and localized error concealed frames obtained using WEC with DPCM bit stream embedding respectively.

Fig. 3.21 shows the plots of the PSNR vs. loss rate of the proposed algorithm along

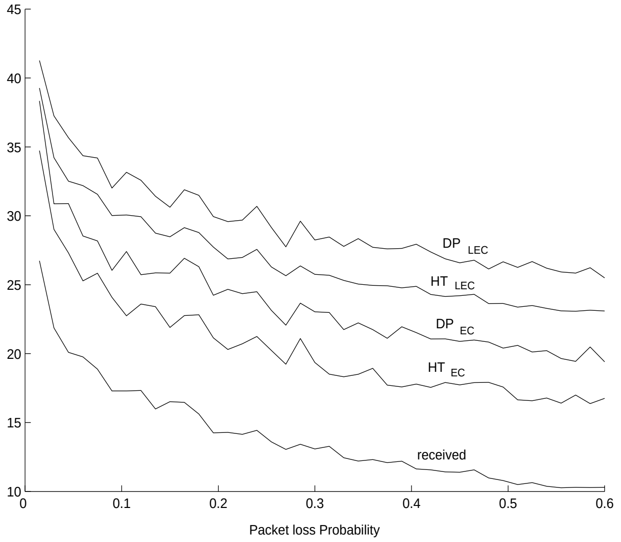


Figure 3.21: PSNR vs. loss rate curves of WEC with halftoning and WEC with DPCM encoding.

with its comparison to WEC with halftone bits embedding. As observed from the figure, both the EC and the LEC versions of the proposed algorithm perform better than WEC with halftone bits embedding. A performance comparison of WEC with DPCM bit stream embedding, WEC with halftone bits embedding, and ECSI is provided in Table. 3.3 for various videos. From the figures and the table, we can conclude that WEC with DPCM bit stream embedding (in the LEC case) outperforms other techniques.

3.9 Adaptive α Modification

In this section, we propose an efficient implementation of the WEC approach where a low-resolution version of a video frame is obtained and embedded in itself at the encoder in the form of watermark data. At the receiver, the watermark is extracted and used as a reference to identify and conceal any packet loss errors [86]. Here, a full-frame DCT is used to embed the watermark in a pixel-wise 2×2 matrix format using spread spectrum watermarking and a correlation-based detector is used to extract the embedded watermark data. Furthermore, a predictive feedback loop is employed at the encoder to estimate the watermark detection accuracy. A detector is added to the encoder and the extracted watermark is predicted. Based on this prediction, the value of the scale factor

α, that determines the strength of the embedded watermark, is modified such that the bit error rate (BER) at the decoder is reduced. Since the encoded makes an "informed" decision on the value of α, we call this informed watermark algorithm [87], [88].

We then propose two variations of this technique with the aim of reducing the perceivable watermarking defects and BER of the detected watermark. In one variation, the sign of the host coefficient is varied based on the sign of the spreading bits. Since we choose mid-frequencies to embed the watermark, the bit-sign modification would make little change to the host video frame. In the second variation, we consider the energy of the embedded signal to be less than a prefixed threshold. This not only reduces the entropy of the embedded signal but also simplifies the detection process.

3.10 Informed WEC Algorithm

In this alternative method for embedding the watermark, the strength of the embedded watermark varied adaptively with the strength of coefficients in which the watermark is embedded based on two aspects; the predicted detector efficiency and sign of the watermark bit. The adaptivity brings in the concept of informed embedding where α is incrementally increased to adapt with the predicted detectability of the watermark at the decoder. This technique highly decreases not only the BER of watermark detection but also the perceivable defects in the video introduced by the watermarking process. Both the aspects of the method are explained in detail herein.

3.10.1 Predicted Watermark Detection

The embedder in this method has an in-built watermark detector which is connected in a feedback loop to adapt to the strength of the embedded signal. The modified watermark embedder is shown in Fig. 3.22. This allows the values of the scale factor α to vary in incremental steps such that the probability of error in detecting the watermark at the receiver is minimized.

The watermarked video frame, after being embedded with a small fixed value of α, is passed on to the matched detector which predicts the value of the extracted watermark. If the embedded watermark bit is extracted correctly, then the value of α is fixed for that

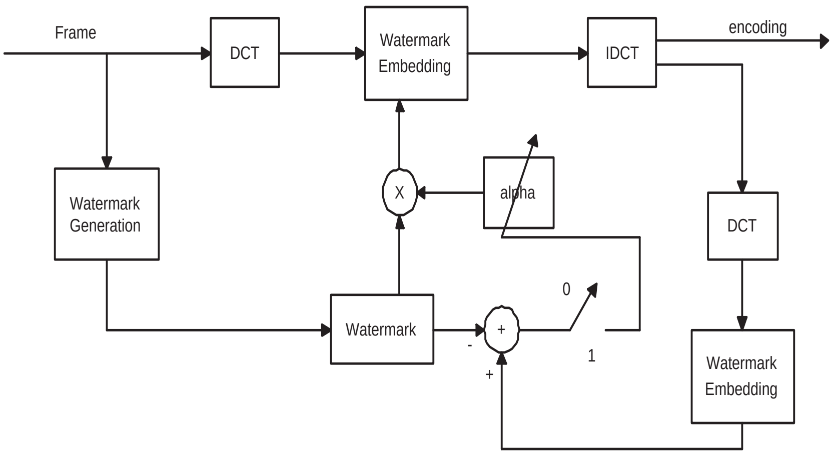


Figure 3.22: Feedback-based watermark embedding model.

bit. However, if the embedded bit is extracted incorrectly, the value of α is increased to a higher value (the next step) and the process is repeated till the watermarked bit is extracted correctly. When it is correctly extracted, the value of α is fixed for that coefficient.

Mathematically, Eq. (3.3) can be modified for the obtaining the embedding equation for the informed watermarking algorithm, and is given by:

$$Y_i(k+\Delta_1, l+\Delta_2) = F_i(k+\Delta_1, l+\Delta_2) + \alpha_i(k,l) \cdot \tilde{w}_i(k,l) \tag{3.15}$$

where $\alpha_i(k,l)$ is the (k,l)-th element of $\boldsymbol{\alpha}_i$, an $\frac{m}{2} \times \frac{n}{2}$ scale factor matrix for the i-th frame. Eq. (3.15) can be alternately written in a matrix form as:

$$\mathbf{Y}_i = \mathbf{F}_i + \boldsymbol{\alpha}_i \cdot \tilde{\mathbf{w}}_i \tag{3.16}$$

where $\tilde{\mathbf{w}}_i$ is defined as in Eq. (3.2).

For the i-th frame, let α_i represent the constant α value in case of non-informed watermarking, and let $\alpha_{i,av}$ represent the average α value in case of informed watermarking algorithm. $\alpha_{i,av}$ is given by

$$\alpha_{i,av} = \frac{4}{mn} \sum_{k=1}^{m/2} \sum_{l=1}^{n/2} \alpha_i(k,l). \tag{3.17}$$

We next state a lemma defining the relation between $\alpha_{i,av}$ and α_i.

Lemma 1. *Given that the probability of error in the detection of the watermark is constant, the average value of the scale factor $\alpha_i(k,l)$ in case of informed watermarking scheme is always less than or equal to the constant α_i value in case of non-informed watermarking scheme for any generic i-th frame in a video sequence, i.e. $\alpha_{i,av} \leq \alpha_i \ \forall i$.*

Proof. The proof of the lemma is based on an assumption that is validated by the empirical data obtained from rigorous experimentation. The experiments initialized $\alpha_i(k,l)$ to a low value for the informed watermarking scheme. The data obtained in all cases showed that 93-95% of the bits were correctly detected with this low "optimum" value of $\alpha_i(k,l)$, say $\alpha_{i,low}$. Therefore, the adaptive change of α_i was performed only in the remaining 5-7% of bits. However, for the non-informed watermarking scheme, α_i was kept constant at a higher value ($\approx 2\alpha_{i,low}$) to obtain the same BER. Based on these results, we deduce that the 93% of $\alpha_i(k,l)$ values are the same. This proves that the matrix $\boldsymbol{\alpha}_i$ is rank deficient.

Now consider the average of this rank deficient matrix. The parameter $\alpha_{i,av}$ in Eq. (3.17) could be rewritten as

$$\alpha_{i,av} = 0.93\alpha_{i,low} + 0.07\alpha_{i,ad}, \tag{3.18}$$

where $\alpha_{i,ad}$ comprise the adaptively varying α values. The first term in Eq. (3.18) is less than $\alpha_{i,low}$. The second term is a product of $\alpha_{i,ad}$ with a small fraction. This implies that even for very high values of $\alpha_{i,ad}$ (approximately 14 times $\alpha_{i,low}$), the second term is less than $\alpha_{i,low}$. Typical high values of $\alpha_{i,ad}$ go up to 4-5 times $\alpha_{i,low}$. Since both the terms are less than $\alpha_{i,low}$, $\alpha_{i,av} < 2\alpha_{i,low}$, and so $\alpha_{i,av} \leq \alpha_i$.

From the lemma, we deduce that the entropy of the embedded signal in case of informed watermarking is lesser than the entropy of the embedded signal when a constant α_i is used. This reduction in entropy implies that we reach a point closer to optimum in the rate-distortion trade-off and therefore a reduction in any perceivable watermarking distortions in the host video frame.

Let us now remove the constraint of keeping BER constant. The error in the detected watermark for the i-th frame, $\mathbf{e}_{i,w}$ can be measured as:

$$\mathbf{e}_{i,w} = |\hat{\mathbf{m}}_i - \mathbf{m}_i| = \mathbf{e}_{i,p} + \mathbf{e}_{i,d} \tag{3.19}$$

where $\mathbf{e}_{i,p}$ is the error due to post-processing operations like inverse-DWT, zooming, etc. and $\mathbf{e}_{i,d}$ is the error due to watermark detection. Since, we are more concerned with the watermark detection error (BER), we specifically define $\mathbf{e}_{i,d}$ as:

$$\mathbf{e}_{i,d} = |\tilde{\mathbf{w}}_{ri} - \tilde{\mathbf{w}}_i| \tag{3.20}$$

where $\tilde{\mathbf{w}}_{ri}$ and $\tilde{\mathbf{w}}_i$ are the extracted and the embedded watermarks respectively. Since both $\tilde{\mathbf{w}}_{ri}$ and $\tilde{\mathbf{w}}_i$ are binary, the error is either 0 (when $\tilde{w}_{ri}(k,l)$ and $\tilde{w}_i(k,l)$ match) or 1 (when $\tilde{w}_{ri}(k,l)$ and $\tilde{w}_i(k,l)$ do not match). When $e_{i,d}(k,l)$ is 1, the switch in Fig. 3.22 is turned on, and a higher value of $\alpha_i(k,l)$ is chosen such that the embedded bits are accurately detected. Thus, the informed watermarking algorithm reduces the BER during the watermark detection process.

The efficiency and the complexity of the $\alpha_i(k,l)$ selection process depends on the proper selection of the initial value of $\alpha_i(k,l)$ for each coefficient. This value is decided based on the strength of each individual coefficient. As will be discussed in Section 5.2.2, a good, close-to-optimum value of $\alpha_i(k,l)$ for a coefficient is obtained by an inverse relationship with its strength. This value empirically proved to be optimum for minimizing BER (extracting the embedded watermark correctly) in most of the embedded bits and a near-optimum in others.

3.10.2 Advantages and Disadvantages

The advantages of this informed WEC technique over the other WEC techniques discussed earlier are threefold: (1) As seen before, the perceivable watermarking defects are negligible due to adaptive scaling of the strength of the embedded bits, (2) the BER values are much less in the informed technique when compared to the non-informed WEC methods, and (3) a higher level of compression can be achieved at the codec as the entropy of the embedded video frame is not as high in case of the informed WEC technique.

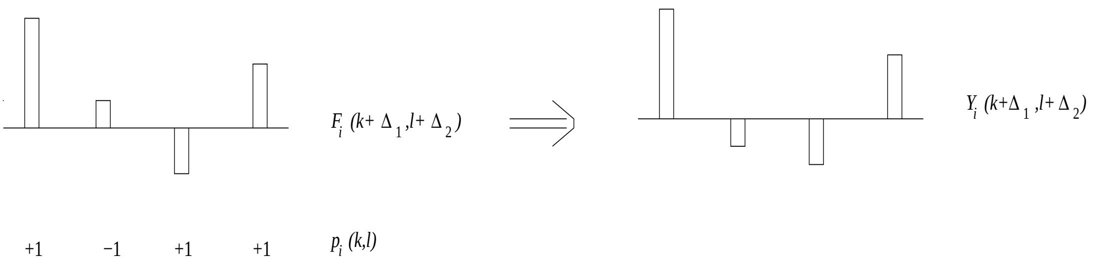

$F_i\ (k+\Delta_1, l+\Delta_2)$
$Y_i\ (k+\Delta_1, l+\Delta_2)$
+1
−1
+1
+1
$p_i\ (k,l)$

Figure 3.23: Illustration of watermark embedding for bit $b = +1$ with bit-sign adaptivity.

As seen from lemma, the informed WEC requires smaller α_i, which implies reduction in the modification of the host coefficients. However, one of the conceivable disadvantages of this technique would be that the complexity of the embedding module will be slightly higher. This nonetheless is more than compensated by the increase in performance at the receiver.

3.10.3 Watermark Bit-Sign Adaptivity

In this variation, the watermark is embedded by adapting the host frame coefficient with the sign of the pseudo-random number. The watermark bit of +1, for example, is embedded by modifying the sign of the coefficient to be in accordance with the sign of $p_i(k,l)$. This could be alternatively considered as a form of multiplicative embedding since the embedded frame $\mathbf{Y}_i$ is obtained by multiplying the host frame $\mathbf{F}_i$ with the sign of embedded bit and that of the pseudo-noise image:

$$\begin{aligned} Y_i(k+\Delta_1, l+\Delta_2) &= F_i(k+\Delta_1, l+\Delta_2) \cdot (\alpha_i(k,l) \cdot \tilde{w}_i(k,l)) \\ &= F_i(k+\Delta_1, l+\Delta_2) \cdot (\alpha_i(k,l) \cdot b \cdot p_i(k,l)). \end{aligned} \quad (3.21)$$

This technique has some unique features that make it appealing for implementation. Apart from working in tandem with the feedback-based informed WEC algorithm, it reduces the parent frame distortion since once the watermark is detected, it can be removed from the host frame by simply multiplying the pseudo-noise matrix with the embedded coefficients. Furthermore, the detection of the embedded bits can be performed based on comparison between the embedded coefficients and the host coefficients at the encoder.

The variation of α_i can again be adjusted to reduce the amount to which the host coefficient set is modified. Along with the informed feedback, if $p_i(k,l)$ and the host coefficients differ substantially, then $\alpha_i(k,l)$ can be reduced to accommodate for the increased modifications. However, since in this technique a minimal set of host coefficients is chosen for modification based on the bit-sign difference, the perceivable watermarking artifacts are minimum.

3.10.4 Watermark Energy Thresholding

In this variation, watermark embedding and extraction is performed based on a pre-fixed threshold for the "energy" in the embedded host coefficients. This is implemented also by considering the sign of the watermark bit. Let the threshold for the energy be T. Then, the criterion for embedding and extraction is:

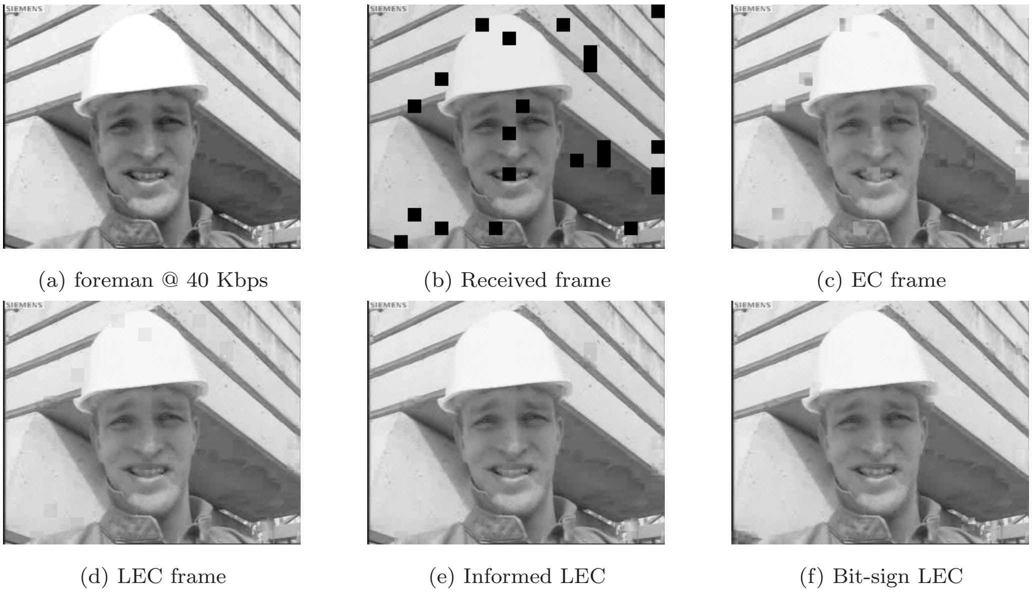
$$b = \begin{cases} +1 & \text{if} \quad E_b > T, \\ -1 & \text{if} \quad E_b \leq T, \end{cases} \tag{3.22}$$

where E_b is the energy in the host coefficients that embed the bit b:

$$E_b = \sum_{k'=2k-1}^{2k} \sum_{l'=2l-1}^{2l} F_i(k' + \Delta_1, l' + \Delta_2) \cdot p_i(k', l') \tag{3.23}$$

Note that the term E_b is not energy of the host coefficients in the true sense. However, it describes the summation of product of signals and is considered here to be a representation of energy. This is explained further in Section 5.2 to be similar to the noise term, and hence minimization of this term is useful.

This technique has a reduction in not only the encoder/decoder complexity but also in the entropy of the embedded coefficients because of its energy-aware embedding and extraction scheme. As seen from Eq. (3.22), the efficiency of this technique depends on the choice of the threshold T. In our experiments, we chose $T = 0$ and used bit-sign adaptivity for the sake of simplicity. Also note that sign of b is a significant factor in dictating the value of E_b.



(a) foreman @ 40 Kbps (b) Received frame (c) EC frame

(d) LEC frame (e) Informed LEC (f) Bit-sign LEC

Figure 3.24: The results for the 36-th frame of the *Foreman* video in CIF resolution compressed to 120 Kbps. The original frame is shown in (a). The received frame is obtained for a mean packet loss probability of 0.12% and variance 0.02%. The PSNR values of the corresponding frames are: (b) 14.2643, (c) 27.6259, (d) 32.4590, (e) 35.9269, and (f) 36.2476. The value of α used in non-informed case was 0.6.

3.11 Simulations with Informed WEC

A sample set of CIF resolution (288 × 352) videos is considered for simulation and the watermark is inserted in the central AC frequencies of the full frame DCT. The *ns-2* simulator is used to generate packet losses with a two-state Gilbert-Elliot Gaussian packet loss model with predefined mean and variance. The packet size was fixed at a macro block (MB) size for a given video transmission and no retransmissions were allowed.

Fig. 3.24 shows the performance of WEC on the *Foreman* video sequence. The value of α was fixed at 0.6 for the non-informed WEC case. Fig. 3.24(b) shows the lossy received frame while Fig. 3.24(c) and (d) show the error concealed (EC) amd locally-scaled error concealed (LEC) frames. Since the watermark is a low resolution version, most of the high frequency information is not retained. Therefore, the reference frame looks “smooth” and the EC frame is “patchy”. This effect is reduced if the EC frame is locally-scaled based on the neighboring luminance values. We call the resulting frame as locally-scaled error concealed (LEC) frame. Fig. 3.24(e) shows the LEC frame with informed WEC and Fig. 3.24(f) represents the LEC frame with bit-sign adaptivity. Here, it should be noted that the bit-sign variation is used on top of informed WEC. In this case, the energy thresholding technique (not shown here due to space restriction) is also performed over informed WEC and resulted in a PSNR= 35.9177 dB.

Table 3.4 shows the performance of the proposed algorithms over various video sequences. As seen, for all the cases except the *Flower* and the *Highway* sequences, the bit-sign adaptive informed WEC gave higher PSNR values of approximately 3 dB over the non-informed WEC technique. In the case of the *Flower* sequence, the energy threshold scheme gave higher PSNR value. We suspect that this is due to the high frequency content in the video.

3.12 Summary

The watermarking technique employed for error concealment is efficiently implemented to protect the low-low wavelet coefficients of the frame from transmission errors by embedding multiple scaled copies of these coefficients in the frame itself. A localized scaling error concealment technique is implemented for improving the perceptual quality of the video frames affected with packet losses. Simulation results with analysis have been presented for various loss rates. Wired to wireless and gray scale to color extensions to the proposed algorithm are presented.

A novel WEC algorithm for video communications is developed where a 2 × 2 set of compact energy coefficients of a video frame are DPCM encoded and embedded into itself using spread spectrum watermarking. The performance of this algorithm is compared to WEC with halftoning and ECSI transmission of DPCM bit stream.

We further presented an informed watermarking algorithm for the application of video error concealment. This WEC approach used full frame DCT to embed a low resolution version of the video frame in itself. The extracted watermark was used for error concealing the lossy received video frame. The algorithm employed a feedback loop to predict beforehand the values of the extracted watermark bits, thereby reducing the overall BER of the detected watermark at the receiver. Bit-sign coefficient modifications and entropy minimization variations were also presented. Based on the obtained results, we

Table 3.4: Performance of the proposed algorithms and variations. PSNR (in dB) for a fixed mean loss 0.15% and variance 0.025%

		Non-informed WEC		Informed WEC		Bit - Sign		Threshold	
Video	Received	EC	LEC	EC	LEC	EC	LEC	EC	LEC
Akiyo	16.1248	28.9114	31.5345	31.9824	34.7662	32.2132	35.5219	32.0145	34.3236
Foreman	14.8279	26.6248	31.4983	32.6751	34.3655	33.1227	35.5787	32.9153	35.4480
Table tennis	15.8202	27.4553	30.1257	31.6767	32.9044	31.9198	35.0286	31.8459	34.1543
Flower	14.5233	26.4553	30.0476	30.8902	32.0967	31.4287	33.3261	31.4076	34.0958
Football	15.0728	27.3546	30.5126	31.0207	33.2548	31.6856	35.3652	31.1362	34.8953
Paris	14.6905	27.0817	30.9213	31.0576	32.9247	31.4889	33.6563	31.3901	33.4156
Tampete	14.4857	27.6707	30.1732	30.5498	33.1634	31.0417	35.2458	30.9093	34.9743
Highway	15.0253	28.7346	32.4018	32.9790	34.3862	33.3252	34.2892	33.2948	33.8266

conclude that the informed watermarking algorithm gave better performance not only in terms of higher PSNR values but also in terms of reduced BER values. The bit-sign variation proved to reduce the perceivable defects introduced by watermarking process, while the energy localization variation provided a more optimal bit rate-distortion trade-off compared to the non-informed WEC method.

A possible future direction would be to identify the areas in the image/video frame where there is a little or no motion and embed the watermark in these areas. This might prove to be useful since viewers often tend to focus on the areas that contain high motion. Another possible future direction would be to devise a watermarking technique that is robust for this application. Spread spectrum watermarking was considered here as a proof of concept due to its ease of implementation in DCT domain. However, it is not ideal for this application due to its large spreading gain and higher entropy. Quantization based watermarking may be an alternative. Ideally, a robust watermarking technique that is targeted for error control applications needs to be developed.

Chapter 4

Watermarked Video Attributes and Implementations

In this chapter, various implementations of WEC algorithms for video are discussed. The implementations could be broadly classified into three different categories, (1) embedding spatial and temporal watermark data in spatial domain, (2) embedding in the temporal domain, and (3) a combined spatio-temporal domain embedding. In the first category, watermark is derived from both spatial and temporal data, i.e., spatial information of the video is derived from the frame and embedded in the frame itself, and motion information of the predicted and bi-directional predicted frames are embedded in the frame. In both these techniques, embedding is performed using the techniques defined in Chapter 3. In the first case, the spatial watermark is generated in one of the two ways: low resolution version of the current I-frame embedded in itself, and the low resolution version of the subsequent P-frames embedded in the current I-frame. All these techniques will be described in this chapter. Note that though there are other error concealment techniques in literature for low bit rate video transmission [5], [89]-[94], the proposed WEC techniques give a better performance and are a logical extensions to Chapter 3.

There are several methods in which the algorithms of Chapter 3 can be extended to video. Due to the facts that the algorithms can be implemented on a frame-by-frame basis and the effects of watermarking on motion vectors is minimal (shown later), this algorithm can be employed in almost all of the currently existing video communication codecs like MPEG-x or H.26x codecs[1]. However, implementing the algorithm on a frame-by-frame basis is not only practically not viable (even though it gives very high PSNR values after reconstruction) as it increases the pre- and post-processing complexity of the video transceivers but also not required as it is possible to hide a lot of information in the video signal without making it noticeable. However, for effectively embedding the watermark spatially in just the key frames, we first need to estimate the effects of watermark embedding on the motion vector information. These effects are identified and analyzed in the next section.

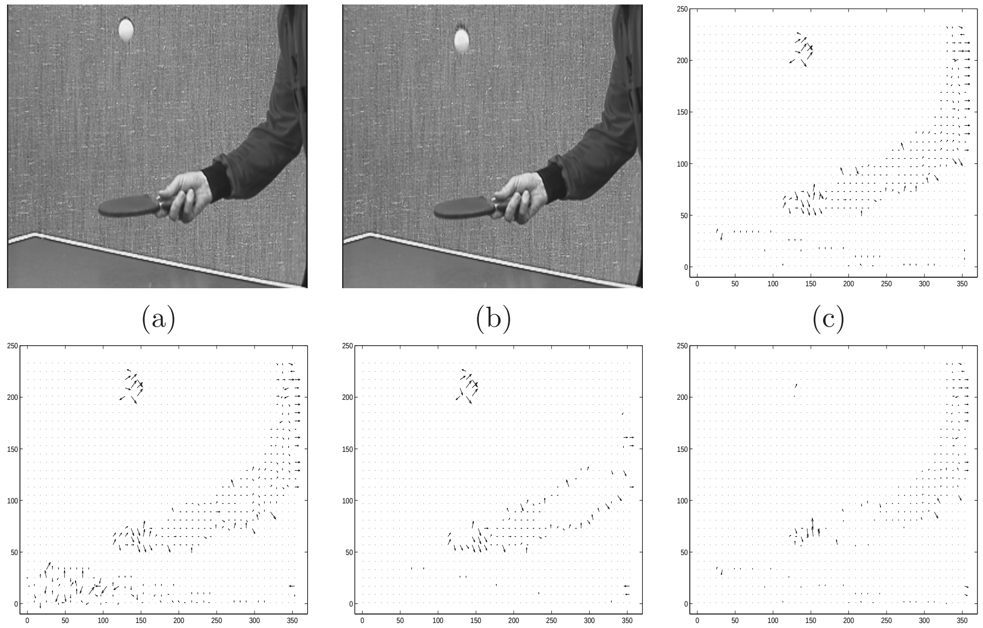
4.1 Effects on Motion Information

We need to consider a trade-off issue with regard to a reduction in the pre- and post-processing complexity against the loss of some efficiency in the error concealment that is provided by the algorithms. We can achieve this trade-off in two ways. The first method deals with spatially embedding the motion information in the DCT domain (which in turn can be implemented in two ways as discussed later in Section 4.3), while the second method involves temporal embedding of spatial information in the motion vectors. The second method is described in Section 4.4.

In the first method, we exploit the motion vector information of the video [95]. As can be seen from Fig. 4.1, the effect of embedding the watermark in consecutive frames does not cause noticeable differences to the motion vectors. Previous work also suggests that any small effects that this process creates can be used to enhance the error concealment at the receiver by coding and transmitting these differences [96]. Hence, we can embed the low resolution versions of the I-frames inside themselves with little change in the motion vector information of the P- and the B-frames provided the P-frames are embedded in themselves.

The effects of the halftone WEC algorithm on motion vectors can be seen in Fig. 4.1. The original k-th and $(k+1)$-th frames of the *Table tennis* video are shown in Fig. 4.1(a) and (b), respectively. Fig. 4.1(c) shows the original motion vectors between these two frames, and Fig. 4.1(d) shows the effect of watermarking on the motion vectors when the

[1] Implementation in H.263 is much more straight forward based on the fact that the algorithm can be extended to work for YUV (4:2:2) and (4:2:0) color video frames.


(a) (b) (c)

(d) (e) (f)

Figure 4.1: (a) Original k-th frame of table tennis sequence, (b) original $(k+1)$-th frame, (c) motion vectors between the above frames, (d) motion vectors when the watermark is embedded in only the k-th frame, (e) motion vectors when the watermark is embedded in both the frames, and (f) difference in motion vectors for (c) and (e).

watermark is embedded only in the k-th frame. Note that these effects are seen in the areas (mid-frequency range of the frame) where the watermark is embedded, specifically in the area of the table in this case. Fig. 4.1(e) shows the effects of watermarking on motion vectors when the watermark is embedded in both frames. This figure shows that embedding in both frames removes the effects of data hiding on motion vectors. Fig. 4.1(f) shows the difference between Figs. 4.1(c) and (e).

The effects on motion vectors may lead to the misconception that we may not be able to embed a watermark in just key frames (I-frames) unless the P-frame already has its own watermark embedded inside it. This however, is not true. Based on the detector operation from Chapter 3, it should be noted that the watermark retrieval process just reads the watermark from the reconstructed image (instead of removing it from the host image) thereby leaving the watermark in the host frames. In fact, this is one of the primary reasons we would want the scale factor α to be at an optimum value so as to minimize the watermarking defects in the host frame. Therefore, if we transmit the modified motion vectors (the ones in Fig. 4.1(d) instead of Fig. 4.1(c) or (e)), we can reconstruct the predicted frames without any motion related errors, even when only the I-frames have the watermarks embedded. This leads to a more optimal point in the complexity-performance trade-off.

4.2 Spatial Embedding of Spatial Information

As explained in Section 4.1, there are two ways of spatially implementing the WEC algorithms: embedding the watermarks generated from the current I-frame in the I-frame itself, and embedding the watermarks generated from the subsequent P-frames in the current I-frame. In the second case, we can have a recursive implementation (looping) where the P-frames that were used to generate the watermark would have the subsequent P-frame embedded in them. Each of these is discussed further on in greater detail.

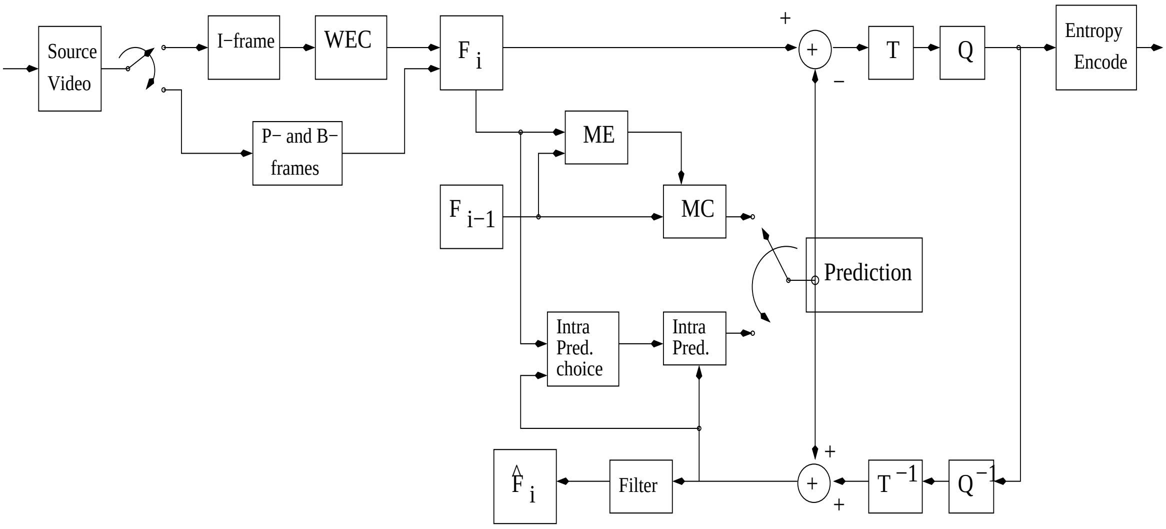
4.2.1 Intra-Coded Frame Embedding in Itself

The block diagram of the embedding process in MPEG-x and ITU-T H.26x encoders is simplified and shown in Fig. 4.2 with the WEC implementation represented on top of the conventional encoder and the decoder. The encoder and the decoder operation shown in Fig. 4.2(a) and (b) are specific to H.264 reference implementation [97]-[100].[2]

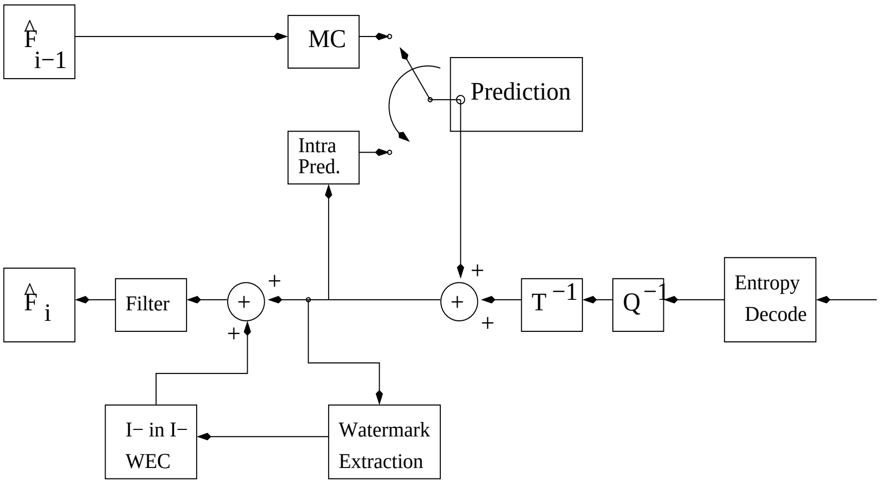
The WEC algorithm implemented in the codecs is that of embedding a reference of I-frame as a watermark inside the I-frame itself. The WEC block in Fig. 4.2(a) represents the watermark generation process using DPCM encoded bitstream discussed in Chapter 3 along with spread spectrum based informed watermarking embedding process. The I-in I- frame WEC block in Fig. 4.3(b) represents the error concealment process similar to the one shown in Fig. 3.19. Note also that the block diagrams represent a codec independent operation, because the implementation of the WEC algorithm is performed as preprocessing and post processing steps in Figs. 4.2(a) and (b), respectively.

We can observe that the WEC algorithm is implemented after the frame is reconstructed from the decoded intra-predicted blocks and in the case of H.264, before the looping deblocking filter. This is done to reduce or remove any blocking artifacts that

[2] The encoder and the decoder operation of H.264 is in accordance with the basic structure of MC-DCT implemented in MPEG-x codecs. In fact, if we exclude the loop (deblocking) filter from the block structures, the resulting operation is similar to that of MPEG-4 [97], [98].



(a) I- in I- WEC implementation in the encoder



(b) I- in I- WEC implementation in the decoder

Figure 4.2: Block diagram of the (a) embedding process and (b) extraction process and WEC implementation for embedding Intra-coded frame reference in itself in H.264 and MPEG-4 codecs.

occur during the error concealment process. Also, the deblocking filter used in H.264 modifies the frame such that the embedded watermark could not be recovered with as low BER as that when it was extracted before the deblocking process.

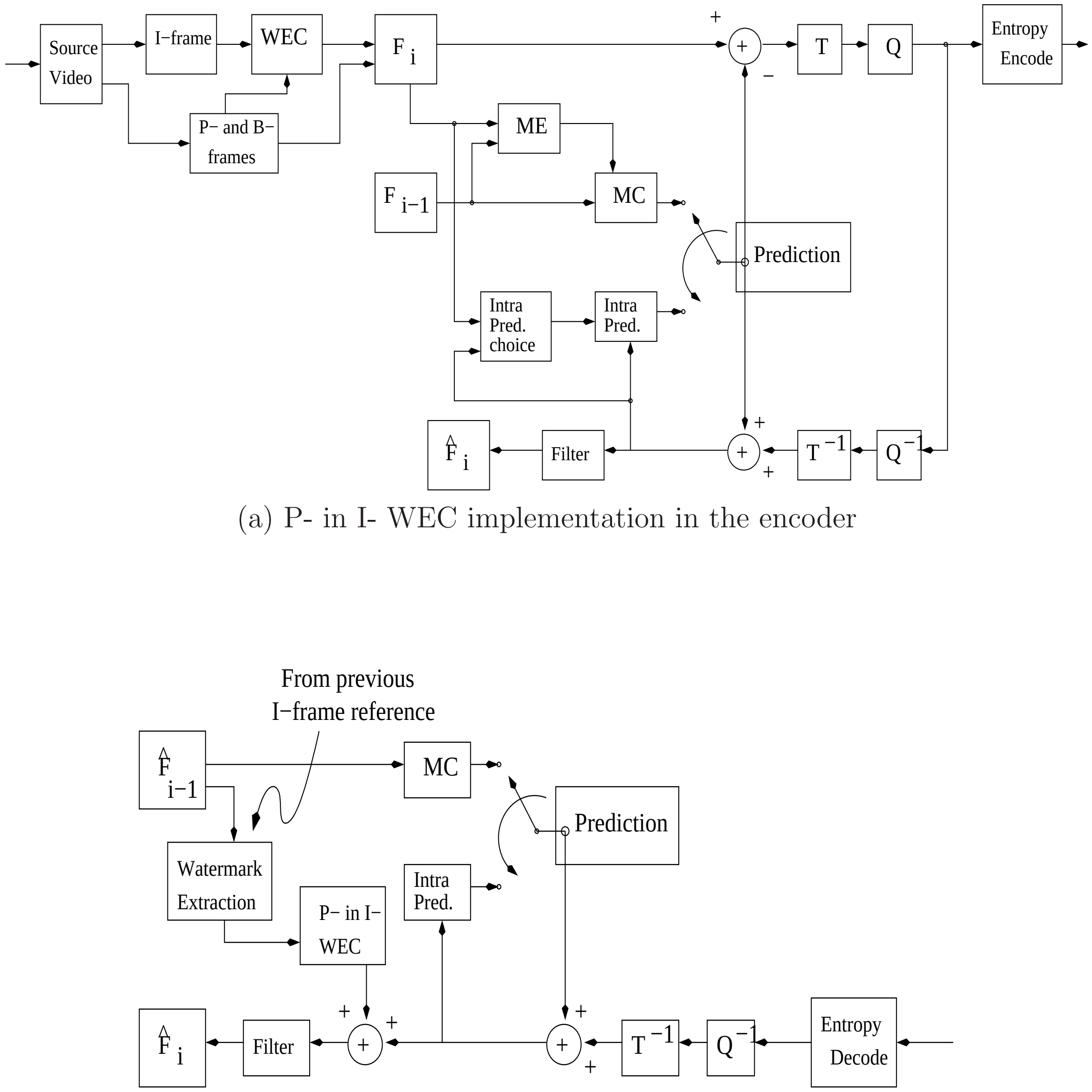
It should be pointed out that this I-frame in the I-frame WEC algorithm can only conceal the spatial errors introduced during the transmission, while the temporal errors continue to be present. Also with this kind of operation, any residual errors that are present in the error concealed I-frame are propagated through the GOP. These temporal errors in the video can be reduced and the error propagation can be easily mitigated by the following simple modification to the I-frame in the I-frame WEC technique.

4.2.2 Embedding Predicted Frame in Intra-Coded Frame

Instead of embedding the I-frames in themselves, we embed the low resolution version of the subsequent P-frame in the current I-frame. This not only enhances the error concealment at the receiver for the current and subsequent P- and B-frames, but also preserves the motion information by giving a reference for the predicted information. For enhanced error concealment in P- and B-frames, their motion vectors can also be embedded along with the low resolution versions of the frames. This technique also mitigates error propagation by error concealing every third frame, thereby creating a high quality reference (both past and future) for the reconstruction of the intermediate B-frames.

The block diagram of the WEC implementation for embedding the subsequent P-frame reference watermark in the current I-frame in conventional codecs is represented in Fig. 4.3. Fig. 4.3(a) shows the codec independent encoder side WEC algorithm implementation. Here, the WEC block includes the watermark generation from the every subsequent 3-rd frame and embedding in the current frame using the same algorithms as discussed in Section 3.10.

Fig. 4.3(b) shows the P-frame in the I-frame WEC algorithm implementation at the decoder side. The operation of error concealment here is a little trickier than in the case of I-frame in the I-frame WEC algorithm implementation. The watermark is extracted from the previously received, reconstructed, and stored third past frame. Since the extracted watermark is that of the current frame, it is then used for error concealment, which



(a) P- in I- WEC implementation in the encoder

(b) P- in I- WEC implementation in the decoder

Figure 4.3: Block diagram of the (a) embedding process and (b) extraction process and WEC implementation for embedding P-frame reference in I-frame in H.264 and MPEG-4 codecs.

is represented by the P- in I- WEC block in Fig. 4.3(b). An implementation issue is synchronization of the third previous frame reference to be stored and extracted for the error concealment of the current frame. However, this is taken care of the conventional MC-DCT decoder where the order of encoding and decoding is fixed to {I,P,B,B,P,B,B...} rather than {I,B,B,P,B,B,P...}.

Apart from the one described in the previous section, there is another reason for introducing the P-frame in the I-frame embedding. The research on visual perception and psychophysical analysis has shown that the subjects consistently rated the constant quality videos higher than the varying quality videos [101], [102]. This means that even though the displayed videos over time had a lesser quality than the maximum achievable quality, but had their quality at a relatively constant level, the subjects preferred these videos to the ones which had variation in quality, even though the quality over time sometimes achieved the maximum achievable quality. For a perceptual evaluation of the quality of a video, PSNR is not a good metric. However, for the lack of a better perceptual measure alternative, we use PSNR for the current analysis.

4.3 Spatial Embedding of Temporal Information

Since the quality of the video is more sensitive to the channel losses that occur on motion information, watermarking acts as a motion vector information recovery tool rather than an alternate way of transmitting the motion vector information. Typically in conventional low bit rate video codecs, the motion compensated error residual (MCER) is encoded and transmitted along with the compressed video data, while the motion vectors are encoded and transmitted as side information. Since the side information is equally prone to channel losses, the encoded motion vector information is not entirely received at the decoder. WEC in this case can provide most of the lost motion information in predicting the P- and B-frames.

The techniques used for spatial embedding are similar to the ones mentioned in the previous section. The temporal information however, is derived from the x and y components of the motion vectors and their predicted difference. The temporal predicted motion vectors are derived from the conventional block matching algorithm that is typically used in the standard MC-DCT based codecs.

Consider the motion vector encoding in the conventional block-based motion estimation algorithms similar to the ones implemented in the video codecs like MPEG-1, MPEG-2 and H.263 [103]. We employ here a full-search block matching motion estimation algorithm that minimizes the mean squared error (MSE) function between the candidate block of size $N \times N$ in the predicted frame over the corresponding reference block in the current frame.

Each macroblock (or sub-macroblock) has a set of motion vectors associated with it. This set varies if the macroblock belonged to a predicted frame or a bi-directional prediction frame. The set of motion vectors can be represented as $MV[t][s](i)$ with the two temporal (forward and backward), two spatial directions (horizontal and vertical), and two displacement precisions (full pixel and half pixel). Here, t and s represent the temporal and the spatial components of the motion vectors, respectively for the i-th macroblock. Typically, $t = 0$ for forward motion vector, $t = 1$ for backward motion vector, $s = 0$ for horizontal component, and $s = 1$ for vertical motion vector component. Note that while for the B-frame, t can be either 0 or 1, t is always 0 for P-frames.

In conventional encoders like MPEG-x or H.26x, due to high correlation between motion vectors of neighboring blocks, they are encoded using DPCM. The predicted motion vectors are defined as:

$$PMV[t][s](i) = \begin{cases} 0 & i = 0 \quad \text{(start of a slice)}, \\ MV[t][s](i\text{-}1) & \text{otherwise.} \end{cases} \tag{4.1}$$

The motion vector difference for the i-th macroblock is $MVD[t][s](i) = MV[t][s](i) - PMV[t][s](i)$ is then encoded using variable length codes (VLCs) and transmitted [100].

However, in our case, for each macroblock, we have a set of four values, the forward and backward motion vector information, and the x and y components of each of the motion vector. We use DPCM encoding as defined in Chapter 3 for encoding these four values. The code is then block ordered. A typical size of this block-ordered matrix is about 6×6 for a 16×16 macroblock. The watermark is formed by concatenating these block ordered matrices of all the macroblocks. It is embedded using the informed watermarking scheme described in Chapter 3. Note that the size of the watermark is just a fraction of the size of the watermark we used for embedding in the case of spatial information.

As in the case of losses occurring during the transmission of side information, any bit errors that occur during the detection of the embedded watermark might lead to noticeable defects in the case of motion vectors. However, neither the channel losses nor the effects of these losses will deteriorate the quality of the extracted watermark and therefore, almost all of the motion vector information could be extracted error free. This motion information could then be used as a reference to recover the lost motion information during the transmission of the side information.

The error concealment of motion information could be performed in one of the two ways: (1) The extracted watermark can serve as a reference for concealing the errors that occur during the transmission of motion vectors, i.e., the lost or erroneously decoded motion vectors could be corrected using the watermark, and these corrected motion vectors could be used to predict the P- and B-frames, or (2) the extracted watermark could be used to predict a P- or a B-frame simultaneously, which will act as a spatial reference for the predicted frames reconstructed using the lossy motion vector information. Although both these techniques perform well with respect to motion vector error concealment, the latter one is computationally more complex than the former, and for this reason, it is a less preferred option.

4.4 Temporal Embedding

In the second method to achieve the processing-efficiency trade-off, we can embed the non I-frames in the motion vectors itself. Previous work suggests that an arbitrary one-dimensional signal can be embedded into the motion vectors of a video and recovered at the receiver without much loss [104]. This can be adapted to the proposed algorithm such that the watermark can now be embedded into the motion vectors and can be recovered without much loss. To reduce the coding overhead in this case, we can now embed every alternate P-frame (instead of every P-frame) in the motion vectors that correspond to the frame. These techniques have not yet been explored completely. We plan to investigate them in greater detail in the near future.

4.5 Volume Embedding using 3D-DCT

A combined spatio-temporal approach is explored here where the watermark data is embedded in the three-dimensional DCT coefficients of the source video. Three variations are used in mark generation and embedding procedures. The first technique employs conventional spread-spectrum embedding in volume data and the second one uses its multiplicative variant. Both these techniques embed the watermark as an 8-bit reference. The third technique modifies the second to least significant bit (LSB) for embedding the watermark bit. We first give a brief explanation of the 3D-DCT coefficients and its properties and then delve into each of these algorithms.

4.5.1 3D-DCT

If the frame resolution is considered to be $m \times n$, as considered in Chapter 3, and if the number of frames being considered for embedding is q (in the temporal dimension), we have a volume coefficient set of $m \times n \times q$. Note that this cuboid with dimensions m, n, and q turns into a cube if $m = n = q$ and into a Euler brick if $m > n > q$. In our case, it is simpler to consider it as q number of $m \times n$ planes since q takes integer values. We calculate the forward and backward 3D-DCT of this cuboid using the following separable transform equations [105]:

$$S(u,v,r) = \gamma(u,v,r) \sum_{t=0}^{q-1} \sum_{y=0}^{n-1} \sum_{x=0}^{m-1} s(x,y,t) \cos(\theta_1) \cos(\theta_2) \cos(\theta_3), \tag{4.2}$$

and

$$s(x,y,t) = \sum_{r=0}^{q-1} \sum_{v=0}^{n-1} \sum_{u=0}^{m-1} \gamma(u,v,r) S(u,v,r) \cos(\theta_1) \cos(\theta_2) \cos(\theta_3), \tag{4.3}$$

where

$$\gamma(u,v,r) = \sqrt{\left[(\frac{2}{m})(\frac{2}{n})(\frac{2}{q}) \right]} C(u)C(v)C(r), \tag{4.4}$$

$$\theta_1 = \frac{(2x+1)u\pi}{2m}, \quad \theta_2 = \frac{(2y+1)v\pi}{2n}, \quad \theta_3 = \frac{(2z+1)r\pi}{2q}, \tag{4.5}$$

and

$$C(k) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } \ k = 0, \\ 1 & \text{otherwise.} \end{cases} \tag{4.6}$$

$S(u, v, r)$ is the corresponding cuboid DCT coefficient of $s(x, y, t)$ in the pixel domain where $s(x, y)$ is the pixel location in frame t and $t \in [0, q-1]$. Note that here $x, u \in \mathbb{Z}^m$, $y, v \in \mathbb{Z}^n$, and $t, r \in \mathbb{Z}^q$, where $\mathbb{Z}$ is the set of positive integers. Thus, we can further divide the DCT cuboid into q planes each of size $m \times n$. Extrapolating the properties of 2D-DCT in the third dimension, we obtain important features that make 3D-DCT attractive for watermark embedding [105]. These include dimensional separability, DCT energy concentration in a specific area, and motion information categorization in the later $m \times n$ blocks (for higher values of q).

4.5.2 8-Bit Reference Data

The energy distribution in the 3D-DCT coefficients in the cuboid is concentrated in the first “quadrant”, i.e., in the region of (u, v, r) where $u \in [0, \frac{m}{2} - 1]$, $v \in [0, \frac{n}{2} - 1]$, and $r \in [0, \frac{q}{2} - 1]$. Therefore, it is feasible to embed huge amount of data in this volume set without introducing any perceivable watermarking defects. We hence choose to embed an 8-bit reference image as watermark.

The reference image is a low resolution version of the reference frame, typically $\frac{1}{16}$-th its size, and is generated by considering a scaled version of the approximation coefficients of its two-level 2D-DWT. The luminance information of each pixel in the reference image is encoded in 8 bits (0, 255). The most significant bit (MSB) is extracted from the 8-bit code of each pixel and a binary image of size $\frac{m}{4} \times \frac{n}{4}$ is formed. Similar binary images are formed with the LSB and all other significant bits. Eight such binary reference images are generated and embedded into $m \times n$ planes with $r \in [0, 7]$.

Embedding this kind of a watermark aids us in two ways: (1) the quality of extracted watermark is very high and therefore the reference will provide higher reconstruction capabilities [106], [107], and (2) complexity of the pre- and post-processing algorithms is reduced due to removal of half-tone and inverse half-tone processes. Furthermore, this choice also enables advantages of (1) using additional compression for the reference in the chrominance domain, and (2) providing reference for inter-coded motion information.

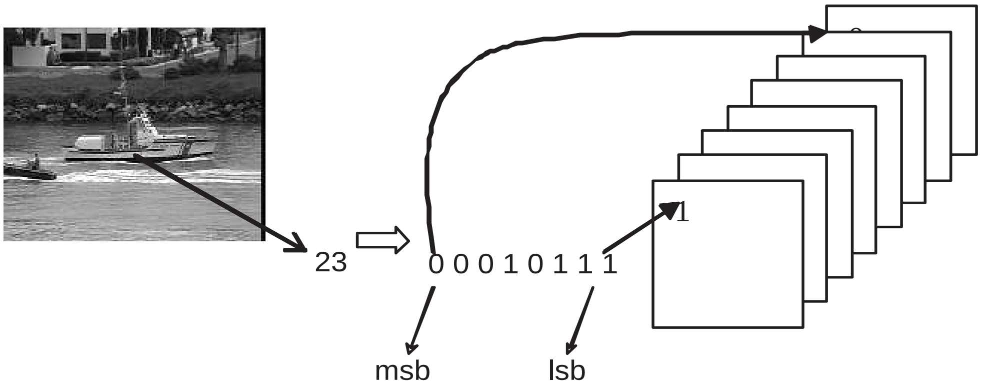


Figure 4.4: The embedding technique.

4.5.3 Embedding in Spatio-Temporal Cuboid

Either the intra-coded frame or one or some of the inter-coded frames can be chosen as the reference frame, as described in the previous subsections. The reference image is the low resolution version of the reference frame. A set of 8 frames is considered for reference embedding. Embedding operation is performed once every group of pictures (GOP). A GOP value of 12 is considered for MPEG-4 and H.264 when the frame rate is 24 fps and a value of 15 when the frame rate is 30 fps.

An 8-bit reference is generated as described in the previous subsection and embedded in the cuboid for $q = 8$. The embedding process is similar to the modified Cox's algorithm described in Chapter 3. Eight binary images of the gray-scale watermark are embedded in the volume data, one image in each plane of size $m \times n$ with $r \in [0, 7]$, such that the binary image generated using the MSBs of the encoded luminance reference is embedded in the plane with $r = 0$ and so on as illustrated in Fig. 4.4. The binary image generated using the LSBs of the encoded luminance reference is therefore embedded in the plane for $r = 7$. Note here that the plane with $r = 0$ corresponds to the DCT coefficients of the intra-coded frame, and every third plane ($r = 3, 6$) to those of the predictive inter-coded frame.

We use three different ways to embed the 8 binary images in the GOP. The first two are variants of the modified Cox's algorithms and given by Eqs. (4.7) and (4.8), respectively. Note that α_i could again be varied based on the strength of each coefficient and the detector performance as in case of the informed watermarking scheme.

$$\text{First :} \qquad Y_r(k+\Delta_1, l+\Delta_2) = F_r(k+\Delta_1, l+\Delta_2) + \alpha_r \cdot \tilde{w}_j(k,l) \qquad (4.7)$$

$$\text{Second :} \qquad Y_r(k+\Delta_1, l+\Delta_2) = F_r(k+\Delta_1, l+\Delta_2)(\alpha_r \cdot \tilde{w}_j(k,l)) \qquad (4.8)$$

where $r \in [0, \mathrm{GOP}]$ and $j \in [0, q-1]$. As mentioned earlier, r is chosen in $[0, q-1]$ for our experimentation. However, there is no constraint on choosing r in the range $[0, \mathrm{GOP}]$. The initial frequency set (Δ_1, Δ_2) is again chosen to be the mid frequencies of $\mathbf{F}_r$ to minimize the perceptual watermarking distortions in the host frame while maintaining robustness to compression. The watermark $\tilde{\mathbf{w}}_j$ is the binary reference image formed by:

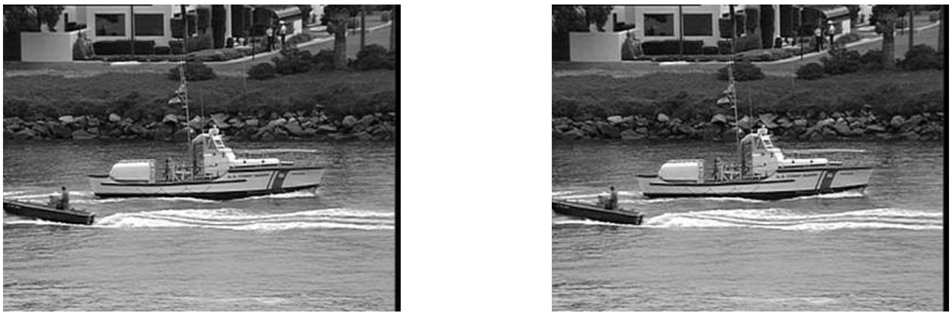
$$\tilde{\mathbf{w}}_j = \breve{\mathbf{w}}_j . * \mathbf{p}_r, \qquad (4.9)$$

where $\breve{\mathbf{w}}_j$ is the pixel repeated binary image formed by j-th significant bit of the 8-bits per pixel (bpp) encoded luminance information of the reference frame and $\mathbf{p}_r$ is the pseudo-noise random image. Note that both $\tilde{\mathbf{b}}_j$ and $\mathbf{p}_r$ are of size $\frac{m}{2} \times \frac{n}{2}$.

Fig. 4.5 shows an example of the embedded frame using the first method for the 58-th frame of the *Coastguard* video sequence. Fig. 4.5(a) shows the original frame without embedding and Fig. 4.5(b) shows the watermark-embedded frame. It can be seen that there are no perceivable watermarking defects in Fig. 4.5(b). The mean PSNR of the luminance channel (Y-PSNR) of the embedded video sequence is 85.7518 dB. The watermark is extracted using a correlation-based detector as explained in Chapter 3. The original and the extracted watermarks are shown in Fig. 4.6.

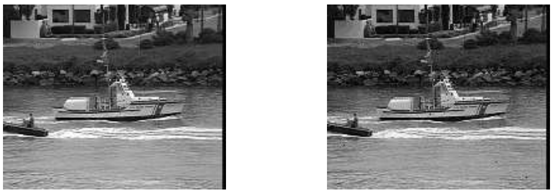
Figs. 4.7(a) and (b) show the extracted watermarks using the first and the second techniques, respectively, for comparison. Both watermarks correspond to the same embedded reference frame. The PSNRs of the extracted watermarks are 78.4958 dB and 72.6311 dB, respectively.

The third technique differs substantially from the first and the second methods. In the third technique, we modify one of the bits in the 8-bpp encoded 3D-DCT coefficients. The selection of this bit is dependent on the trade-off between the quality of the extracted watermark, the perceptual distortion created in the host frame, and the robustness of the embedded watermark to channel errors. Based on our application of error concealment,



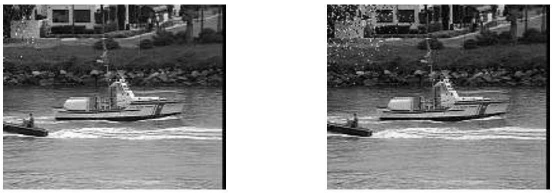
(a) (b)

Figure 4.5: (a) The 58-th frame of the *Coastguard* sequence before watermark embedding, and (b) after watermark embedding.



(a) (b)

Figure 4.6: (a) The original watermark, and (b) the extracted watermark, PSNR = 77.1030 dB.



(a) (b)

Figure 4.7: The extracted watermarks for (a) the first technique, and (b) the second technique.

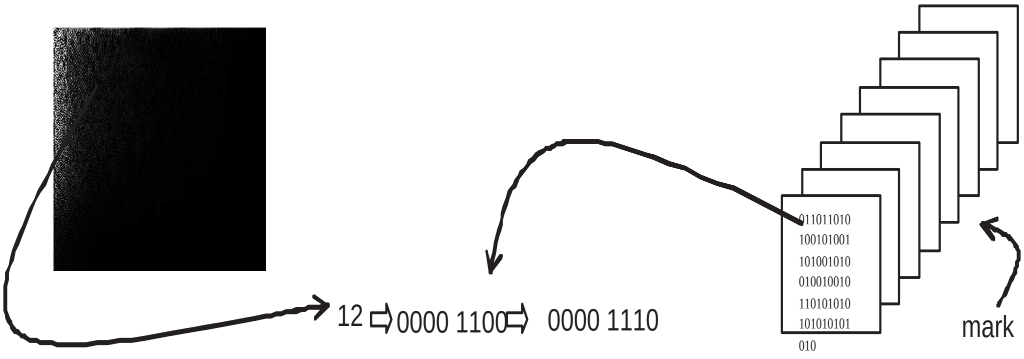


Figure 4.8: The third embedding technique.

we choose this bit to be the second LSB, i.e., $j = 6$, the 7-th bit.

The modification of the second LSB is a replacement process i.e., for each 8-bpp encoded 3D-DCT coefficient (k, l) of the luminance information of the host plane F_r, b_6, is replaced with $w_j(k, l)$. This can be represented as:

$$\text{Third}: \qquad Y_{r,b}(k,l) = F_{r,b}(k,l) \oplus w_j(k,l) \tag{4.10}$$

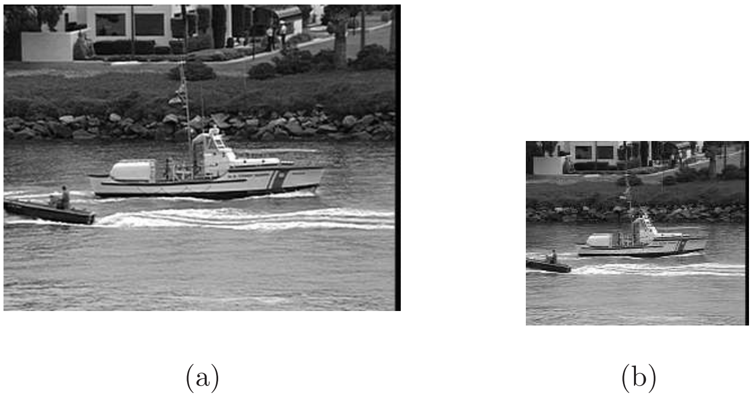
where $\oplus$ represents the bit replacement operation that changes the 7-th bit of the 8-bit code for the (k, l)-th pixel. The subscript b represents the bit-change operation. Fig. 4.8 illustrates the operation of the third method.

Figs. 4.9(a) and (b) show the 58-th frame of the *Coastguard* video sequence after embedding and the extracted watermark, respectively. The average Y-PSNR of the embedded sequence is 93.1648 dB and the PSNR of the extracted watermark is 71.3395 dB. From the PSNR values, we observe that the extracted watermark has highest quality if it is embedded using the first technique.

4.5.4 Advantages and Disadvantages

In this section, we discuss the advantages and the disadvantages of volume embedding over the regular 2D embedding. We then bring out the positive and negative aspects of each of the embedding algorithms that we have outlined.

There are two clear advantages of employing the volume embedding using the 3D-DCT over the conventional 2D-DCT approaches: (1) More information can be embedded in the



(a) (b)

Figure 4.9: (a) The watermark embedded frame for the third technique, and (b) the extracted watermark.

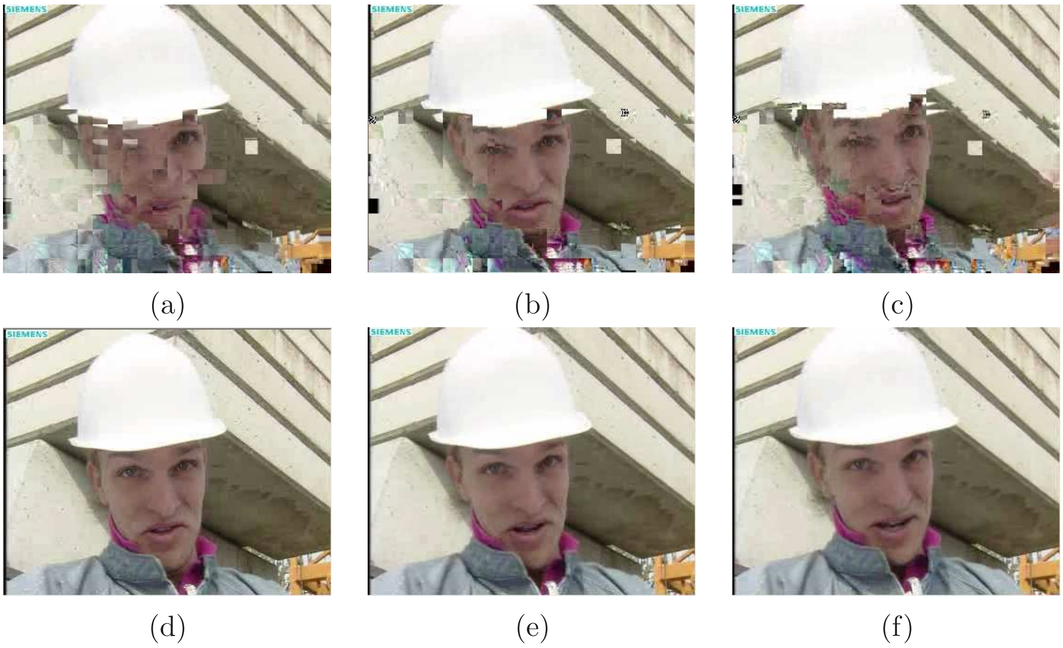
volume data due to just the sheer number of coefficients available to embed [106], [107], and (2) The extracted watermark quality is higher due to the embedding of 8-bit encoded reference image in our case. However, we can incur other advantages based on these two. For instance, we can increase the robustness of the embedded reference by further encoding it without worrying about the amount of increase in the bit rate of the encoded watermark. The encoded watermark will be more robust to channel attacks. Another added advantage would be that this process of volume embedding would allow enough leeway for encoding and embedding sufficient amount of chrominance information, if that is a requirement. Furthermore, we would reduce the computational complexity of the system due to removal of half-tone and inverse half-tone processes. However, this reduction is contingent on the computation of the 3D-DCT. If this operation is performed prior to compression, then the reduction in computational complexity is not much, and we may not even not notice an improvement. If however, the computation is done while compression (since the 3D-DCT is separable, the transformation in the third dimension is just an added one-dimensional DCT operation), we may be able to notice some gains both in computational complexity and quality of extracted watermark.

The three techniques we used for embedding in the volume cuboid differ in more than one ways. Even though the first two techniques are variants of the modified spread spectrum algorithm, the first one is an addition based technique and therefore, by choosing a sufficiently low value of α, we can reduce the effect of the host coefficient modification.

In the second technique which is multiplicative, the value of α which is dependent on the inverse of the strength of the host coefficient, would produce a change in the coefficient that might introduce certain perceivable watermarking artifacts. Therefore, the second technique will not only disturb the host frame, but also reduce the quality of the extracted watermark. However, embedding the watermark is much easier, faster and less complex when compared to the first technique as the process just involves changing the sign of the host coefficient based on the sign of the spreading bit. The third technique involves the replacement of one of the bits (the 7-th in the 8-bpp code of the host coefficient). It therefore is similar to the quantization index modulation (QIM) based methods with a predefined quantization step. Note that the increased capacity of volume embedding makes QIM and QIM-like techniques feasible. Since we change the 7-th bit of the 8-bit code, the quantization step varies in the range of $[0, 2]$. This technique therefore inherits the advantages and disadvantages of the QIM based techniques over spread spectrum based techniques, including the facts that it is more seamless producing lesser perceivable watermarking artifacts and it is more robust to channel effects. This technique also has the advantage of reduced complexity as it involves modifying just one bit. However, as mentioned before, this fact needs to be considered with a bit of salt as it lies on top of an already complex 3D-DCT operation.

4.6 Simulations

In the case of video implementations, though the algorithms were tested for CIF, QCIF, SIF, progressive SDTV (VGA), and 320 × 240 resolutions (QVGA), with no compression, low, medium, and high compression, using H.264, MPEG-4, and MPEG-2 compression codecs, with varying loss rates, the results presented here are specific to CIF video sequences (the sequence is fixed to *Foreman* to keep the consistency in comparing performances) for a compression of 120 kbps using both H.264 and MPEG-4 for varying packet losses. The performances of the I-frame in the I-frame WEC and P-frame in the I-frame WEC algorithms are first presented and then a comparison of these is performed for varying packet losses. These are followed by sample results of the first 3D-DCT WEC algorithm implementation.



(a) (b) (c)

(d) (e) (f)

Figure 4.10: Frames of the *Foreman* video processed using MPEG-4: (a) MPEG-4 error concealment for a packet loss of 0.3%, Avg. Y-PSNR = 22.89 dB, frame 120, (b) frame 121, (c) frame 122, (d) I- in I-frame WEC implementation, Avg. Y-PSNR = 41.58 dB, frame 120, (e) frame 121, and (f) frame 122.

4.6.1 Spatial Embedding

Fig. 4.10 shows the sample results of the I-frame in the I-frame WEC algorithm's performance when implemented on top of the conventional MPEG-4 codec. Figs. 4.10(a), (b), and (c) represent the frames 120-122 of the *Foreman* video with basic error concealment in MPEG-4 when used in the advanced simple (AS) profile [108] for a packet loss percentage of 0.3, while Figs. 4.10(d), (e), and (f) represent the same set of frames with the I-frame in the I-frame WEC algorithm implemented on top of (a), (b), and (c), respectively. Note that the WEC implementation not only does a good job at reducing and removing errors (approximately 18 dB improvement in the average Y-PSNR values), but also effectively mitigates the propagation of these errors through the subsequent frames (an issue typically observed in the MC-DCT based encoders).

Fig. 4.11 shows the sample results for the subsequent P-frame embedding in the current I-frame WEC technique implemented on top of the conventional H.264 codec. Figs. 4.11(a), (b), and (c) represent the frames 127-129 of the *Foreman* video with main profile H.264 error concealment for a packet loss percentage of 0.3. Figs. 4.11(d), (e),

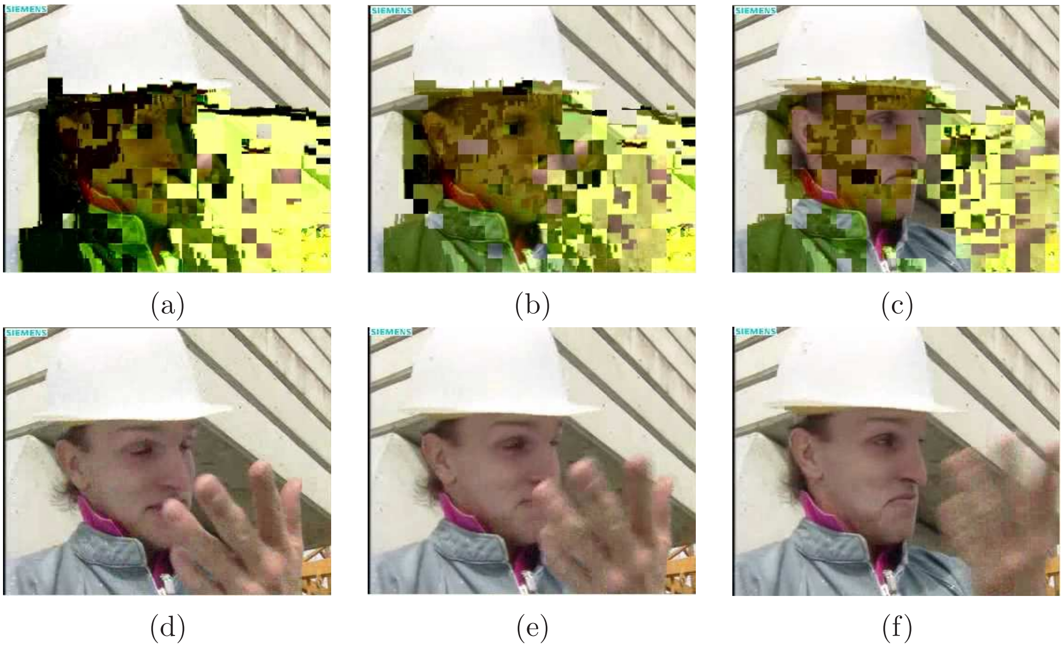
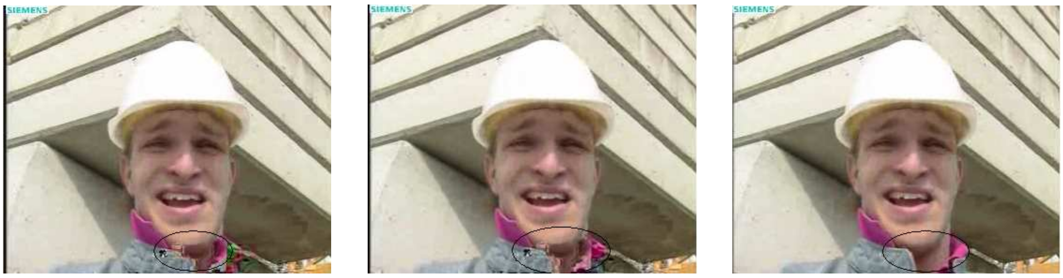


Figure 4.11: Frames of the *Foreman* video processed using H.264: (a) H.264 error concealment for a packet loss of 0.3%, Avg. Y-PSNR = 20.48 dB, frame 127, (b) frame 128, (c) frame 129, (d) P- in I-frame WEC implementation, Avg. Y-PSNR = 42.23 dB, frame 127, (e) frame 128, and (f) frame 129.

and (f) represent the same set of frames error concealed using the P-frame in the I-frame WEC algorithm implemented on top of (a), (b), and (c), respectively.

Note that for the same packet loss percentage, the amount of loss occurring in the H.264 video is more than the amount of loss in MPEG-4 processed video sequences. This is as expected because H.264 is built for the purpose of efficient compression and to withstand compression artifacts even at lower compression bit rates. However, MPEG-4 is built for the scalability and is tailored to cater for multiple applications. For this reason, it has many more profiles (17) when compared to the 3 profiles that exist in H.264. It is this scalability that makes MPEG-4 more robust to transmission errors rather than compression errors.

Also note that though the errors that occur due to the same packet loss percentage are more in the case of H.264 processed video when compared to the MPEG-4 videos, the property of mitigating this error propagation is more in H.264, i.e., the variation in the amount of loss that occurred in Figs. 4.10(a) through (c) is less than the variation in the amount of loss in Figs. 4.11(a) through (c). This increase in variation of the amount



(a) (b) (c)

Figure 4.12: Difference in the two WEC algorithm implementations highlighted for *Foreman* video, frame 15, processed using H.264 for a packet loss of 0.6%: (a) No WEC, (b) I- in I-frame WEC, (c) P- in I-frame WEC.

of loss comes from H.264's efficient compression capabilities.

In the quality-performance comparison of the two WEC algorithm implementations in Figs. 4.10(d) through (f) and Figs. 4.11(d) through (f), we notice an enhancement in the performance in the case of P-frame in the I-frame embedding. This is not only due to the fact that the P-frame in the I-frame gives approximately the same objective quality values after the WEC algorithm implementation on the packet loss affected video as the I-frame in the I-frame WEC algorithm does (as seen in the figures), but also reduces any errors that occur only in P- and B-frames that the latter algorithm does not conceal. This aspect can be more closely observed by looking at a set of frames in a different GOP which has errors only in P- and B-frames.

Fig. 4.12(a) shows the 15-th frame of the *Foreman* video sequence with no WEC algorithm implementation on top of H.264 codec and a packet loss percentage of 0.6. It is a predictive coded frame with the error that occurred in the bottom of the frame (the erroneous area is circled). Fig. 4.12(b) shows the same video frame with the I-frame in the I-frame WEC algorithm implementation on top of conventional H.264 implementation. Again the error area is circled, and we observe that the error has not been concealed. This is as expected for the reasons explained in Section 4.2.1. However, the P-frame in the I-frame WEC algorithm implementation removes (conceals) the error as seen in the same circled area of Fig. 4.12(c). Therefore, the P-frame in the I-frame WEC algorithm achieves a higher performance when compared to the I-frame in the I-frame WEC algorithm.

Table 4.1: Performance of the proposed algorithms in case of compression with MPEG-4 codec. PSNR (in dB) values are presented for a fixed mean packet loss 0.20% and variance 0.024%

	No compression			360 Kbps		
Video	No WEC	I in I	P in I	No WEC	I in I	P in I
Akiyo	25.52	44.13	44.39	18.08	32.53	35.87
Foreman	24.83	42.12	43.66	17.65	31.02	34.73
Table tennis	26.11	44.06	44.48	18.42	33.35	33.51
Flower	23.57	42.23	42.94	16.91	32.44	33.89
Football	24.63	43.34	43.59	17.55	33.38	33.16
Paris	24.06	42.32	43.61	17.11	31.84	34.14
Tampete	25.27	43.96	44.01	17.80	33.68	35.42
Highway	26.33	44.29	44.74	16.67	33.95	34.86

Table 4.2: Performance of the proposed algorithms in case of compression with H.264 codec. PSNR (in dB) values are presented for a fixed mean packet loss 0.20% and variance 0.025%

	No compression			360 Kbps		
Video	No WEC	I in I	P in I	No WEC	I in I	P in I
Akiyo	25.87	43.48	44.54	17.28	32.43	34.89
Foreman	23.58	42.06	42.79	15.92	31.42	33.68
Table tennis	26.48	43.07	43.84	16.55	32.62	34.03
Flower	22.76	41.68	42.89	14.96	31.72	33.19
Football	23.88	43.37	44.28	16.01	32.49	34.12
Paris	24.54	42.78	43.63	17.16	33.44	34.29
Tampete	25.06	43.36	43.74	17.86	32.87	33.92
Highway	23.37	43.61	43.66	16.47	32.85	34.73

Table 4.1 shows the performance of the I- in the I-frame and the P- in the I-frame WEC algorithm implementations for a fixed loss of approximately 0.19% in the cases of no compression and a compression of 360 Kbps with MPEG-4 codec implementation. As seen from the table, in almost all the cases, the performance of the P-frame in I-frame algorithm is the best, i.e., it gives the highest average Y-PSNR values, and more importantly, closer to the constant perceptual quality performance.

Table 4.2 shows the performance of the WEC techniques implemented over H.264 codec for no compression and a bit rate of 360 Kbps compression for a fixed mean packet loss percentage. As seen from the table, the performance of the P-frame in the I-frame

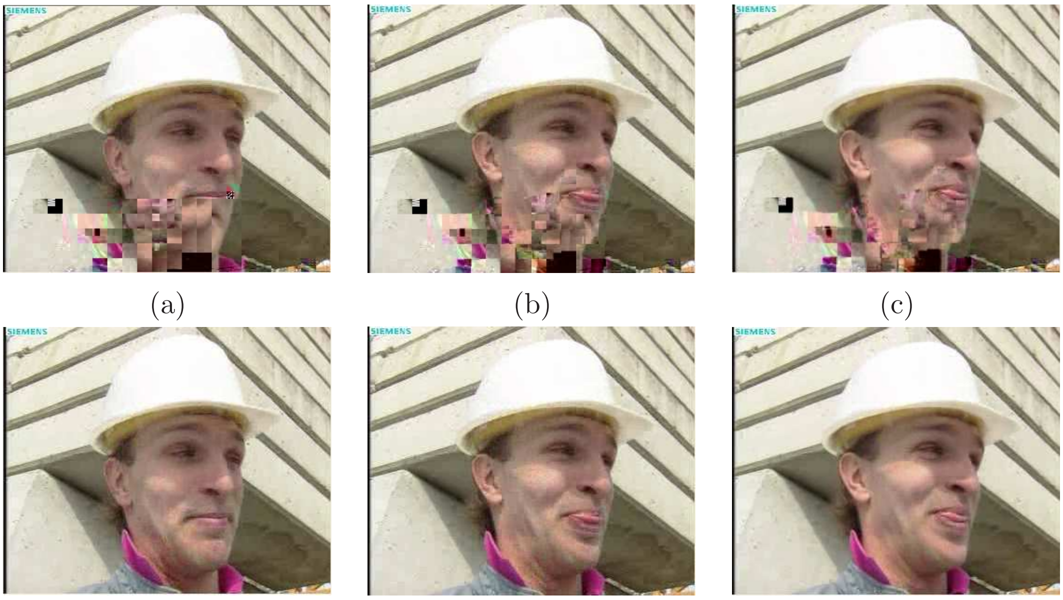
WEC algorithm supersedes the I-frame in I-frame WEC algorithm and the conventional codec error concealment (no WEC implementation) even for this codec. However, it should be pointed out that the performance of WEC algorithms in general can be observed to decrease with increasing amount of compression. This is typical to what is expected as the scale factor in the quantization process will corrupt the embedded watermark data thereby resulting in a lower quality reference at the receiver, as explained in Chapter 3.

4.6.2 Volume Embedding

In the case of 3D-DCT volume embedding, the algorithm described by Eq. (4.7) is implemented in the first 8 planes of the DCT cuboid of size 352 × 288 × 12 (12 frames in the GOP). The selection of the mid frequency set of coefficients is different for each plane. This implementation is used for the initial testing phase. However, we believe that the usage of frames 2 through 10 (mid temporal frequencies) would be a better choice when compared to implementation in the first 8 frames.[3]

Fig. 4.13 shows the performance of the first technique described in Section 4.5.3. Figs. 4.13(a), (b), and (c) represent frames 3, 4, and 5 of the *Foreman* video respectively for a packet loss percentage of 0.2 when compressed and transmitted using MPEG-4 video codec. Figs. 4.13 (d), (e), and (f) represent the same set of frames corresponding to (a), (b), and (c), respectively, but with the 3D-DCT WEC algorithm implementation. Note that even though the watermark is the reference of just the I-frame, since an 8-bit reference is used, the quality of the extracted watermark is very high and the WEC algorithm is efficient in effectively removing the I-frame errors and mitigating the P- and B-frame error propagations and therefore would perform as well as the P-frame in the I-frame WEC algorithm. This however, does not give it the capability of removing or reducing independent P- and B-frame errors. Reducing these types of errors would involve either embedding multiple references (both for the I- and P-frames) in the 3D-DCT volume cuboid, or intelligently selecting either the I-frame or the P-frame for the

[3] Since the normalized differences in the temporal frequencies represent the amount of motion in the GOP, the low temporal frequencies happen to be the ideal choice. However, embedding the reference in the low temporal frequencies not only tends to distort the motion vectors significantly but also incurs higher amount of error due to error 1propagation down the GOP, thereby making it difficult to extract the watermark with superior quality. We therefore opt for the mid temporal frequencies as a reasonable trade-off.



(a) (b) (c)

(d) (e) (f)

Figure 4.13: 3D-DCT WEC implementation for *Foreman* video with packet loss of 0.2% processed using MPEG-4: (a) No WEC, frame 3, (b) frame 4, (c) frame 5, (d) 3D-DCT WEC, frame 3, (e) frame 4, and (f) frame 5.

reference watermark in every GOP.

Table 4.3 shows the performance of the three methods discussed in Section 4.5.3 given by Eqs. (4.7), (4.8), and (4.10) along with the performance of no WEC algorithm implementation in the cases where there is no compression and 360 Kbps compression with MPEG-4 codec. The performance values, measured by average Y-PSNR (in dB) for the video sequences, are for the fixed packet loss percentage (mean) of approximately 0.19%. We can observe from the table that the performance of the first method is the best, followed by that of method 3 and method 2, in that order. Method 3 gives a better performance than method 2 due to its little to no change of the host video DCT coefficients when compared to method 2.

Table 4.4 compares the performance of the three 3D-DCT techniques in the case of no compression and a low bit rate compression level of 360 Kbps using the H.264 codec. In both tables, the WEC techniques are implemented on top of the codecs' implementation. Though the improvement in performance is reflected to be the best in method 1, method 3 comes a close second even for H.264. We can also observe that as the amount of compression increases (the compression bit rates decreases), the performance

Table 4.3: Performance of the 3D-DCT algorithms in the case of compression with MPEG-4 codec. PSNR (in dB) values for a fixed mean loss 0.19% and variance 0.023%

	No compression				360 Kbps			
Video	No WEC	Method 1	Method 2	Method 3	No WEC	Method 1	Method 2	Method 3
Akiyo	29.42	49.37	44.53	48.76	18.38	34.02	29.16	32.64
Foreman	28.93	48.52	43.66	47.86	17.74	33.65	27.89	31.96
Table tennis	29.18	49.51	44.61	48.11	17.97	34.39	28.13	32.78
Flower	27.62	46.89	42.04	45.71	16.38	33.67	27.57	32.09
Football	28.91	48.76	44.29	47.32	17.88	33.87	28.69	32.97
Paris	27.90	47.13	42.87	45.64	16.89	33.92	28.02	31.86
Tampete	29.32	49.48	44.62	48.03	19.01	35.17	30.11	33.58
Highway	28.67	48.64	43.87	47.42	17.38	33.85	28.34	32.68

Table 4.4: Performance of the 3D-DCT algorithms in the case of compression with H.264 codec. PSNR (in dB) values for a fixed mean loss 0.19% and variance 0.022%

	No compression				360 Kbps			
Video	No WEC	Method 1	Method 2	Method 3	No WEC	Method 1	Method 2	Method 3
Akiyo	28.39	48.07	41.28	45.92	16.45	32.78	27.54	30.96
Foreman	27.92	47.63	41.59	45.56	16.03	32.14	27.22	30.27
Table tennis	28.30	47.94	41.52	46.05	16.21	32.91	27.18	31.24
Flower	26.89	45.78	40.36	43.94	14.96	31.84	26.55	30.28
Football	27.67	47.58	42.62	46.04	15.93	32.26	27.38	30.54
Paris	27.33	47.06	41.47	45.14	15.98	31.86	27.23	30.41
Tampete	28.51	48.32	42.44	46.37	16.62	33.12	27.95	32.56
Highway	27.68	47.83	41.78	45.86	15.79	32.54	27.12	30.87

of the WEC techniques decreases. This decrease is more so in the case of the spatio-temporal technique. From the Tables. 4.3 and 4.4, we observe that in the case of higher compression, 3D-DCT is not as robust as P- in the I- or I- in the I- WEC algorithm implementations.

4.7 Summary

The two dimensional and the three dimensional techniques discussed here give excellent performance enhancements when compared to other existing error concealment techniques not only with regard to their efficient implementation based on the properties of the video but also with regard to end user perceptual video quality. We first discuss various 2D WEC algorithm implementations both in case of spatial (I-frame in the I-frame and P-frame in the I-frame), temporal in spatial, and temporal schemes. We then discuss and test experimentally the performance of the 3D combined spatio-temporal WEC algorithm. The performance of the proposed implementations of the WEC algorithms is brought out by the simulations presented. Comparison across methods reveals that in the case of spatial embedding, the P-frame reference embedding in the I-frame WEC algorithm implementation gives the best performance (highest average Y-PSNR values), while in the case of 3D-DCT spatio-temporal embedding, the first method of additive watermark embedding gives the best performance. Moreover, the P-frame in the I-frame WEC algorithm is suggested to show a constant perceptual quality behavior.

Chapter 5

Information Theoretic Framework for WEC

In this chapter, an information theoretic approach to the proposed WEC algorithms is introduced and analyzed. To proceed with this framework, analysis of the embedder and the detector performance is the key. Assuming that the embedded data is transmitted as the side information, the framework is built on minimizing the bit detection error rate (BER). Estimation of BER in correctly reading the embedded watermark leads to proper estimation of the overall distortion in the reconstructed video signal and thereby in the error concealed video. Therefore, we first deal with WEC detector performance analysis in the following sections and then discuss its effects using the conventional information theoretic approach.

5.1 Background

The application to data hiding to the field of image and video communications is novel and little investigation has been done so far to develop efficient algorithms. An effective WEC algorithm which embeds a low resolution version of the image in itself was proposed in Chapter 3 [86] and extended to video in Chapter 4. Here, a full frame DCT is used to embed the watermark in a pixel-wise 2×2 matrix format using spread spectrum watermarking and a correlation-based detector is used to extract the embedded watermark data.

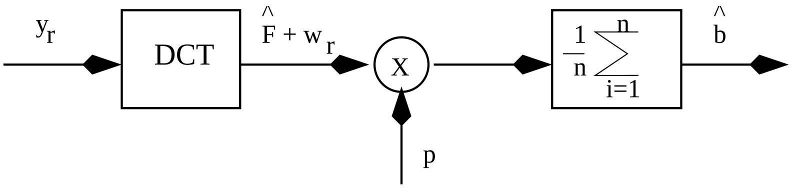


Figure 5.1: Watermark detection model

Evaluation of the detector performance in conventional data hiding algorithms is challenging to model due to the inherent randomness in the distribution of the image DCT coefficients. In this work, we attempt to mathematically evaluate the performance of the detector in the WEC algorithm proposed in [86].

The performance evaluation of block-based correlation-like detectors has been documented in the literature [109]-[111]. The block-based image DCT coefficients have been modelled to have a generalized Gaussian distribution (GGD) in [109]. An optimum watermark detector is sought in [110] and [111] using the maximum likelihood estimation for the algorithms proposed here.

However, the analysis provided in the literature so far is not applicable to our method of WEC due to two reasons: (1) the analysis is developed for block-based DCT embedding systems, whereas, we use full frame DCT embedding, and (2) we vary the strength of the watermark for each coefficient we embed. We therefore provide a performance comparison, both analytically and using simulations, for block-based and full frame DCT embedding approaches.

5.2 Detector Performance Analysis

The detector model is shown in Fig. 5.1. In general, the DCT coefficients of an image which is transformed using 8 × 8 blocks can be modelled to follow a Gaussian or a Laplacian [112] distribution. In later work that followed [113], a GGD for which both Gaussian and Laplacian are special cases, was shown to be a more accurate model of the block-based DCT coefficients. In our work however, we use a full frame DCT. Since it is difficult to model full frame DCT AC coefficients, we extrapolate the modelling of 8 × 8 block AC coefficients to that of a select set of full frame AC coefficients.

The GGD is of the form:

$$f(x) = \frac{c\beta(c)}{2\sigma\Gamma(1/c)} \cdot exp\left\{-\left(\beta(c)|\frac{x}{\sigma}|\right)^c\right\}, \tag{5.1}$$

with

$$\beta(c) = \sqrt{\frac{\Gamma(3/c)}{\Gamma(1/c)}}, \tag{5.2}$$

where $\Gamma(\cdot)$ represents a Gamma function, c is the shape parameter, and σ is the standard deviation of the GGD. $f(x)$ turns into Laplacian density if $c = 1$ and into Gaussian density if $c = 2$. We know that $\mathbf{p}$, a zero mean unit variance pseudo-random noise matrix, has a Gaussian distribution and we assume that $\mathbf{Y}_r$, the DCT coefficient matrix of the received frame, follows a GGD. If we set $\mathbf{Y}_r = \hat{\mathbf{Y}}$, an estimate of the embedded image, then from Eqs. (3.3) and (3.5), we have

$$\begin{aligned}
\lambda(k,l) &= \sum_{k'=2k-1}^{2k} \sum_{l'=2l-1}^{2l} \hat{Y}(k'+\Delta_1, l'+\Delta_2) \cdot p(k',l') \\
&= \sum_{k'=2k-1}^{2k} \sum_{l'=2l-1}^{2l} \hat{F}(k'+\Delta_1, l'+\Delta_2) \cdot p(k',l') \\
&\quad + \alpha \cdot \hat{b}_i \cdot \sum_{k'=2k-1}^{2k} \sum_{l'=2l-1}^{2l} p^2(k',l') \qquad (5.3) \\
&= \lambda_n(k,l) + \alpha \cdot \hat{b} \cdot \lambda_s(k,l), \qquad (5.4)
\end{aligned}$$

where $\hat{F}(\cdot,\cdot)$ has a GGD and $\hat{b}$ is the estimation of the embedded bit and is given by

$$\hat{\hat{w}}(k,l) = \hat{b} \cdot p(k,l). \tag{5.5}$$

In Eq. (5.3), $\lambda_s(k,l)$ represents the signal part and is a product of two Gaussian random variables, whereas, $\lambda_n(k,l)$ represents the noise part and is a product of a GGD and a Gaussian random variables, which is difficult to compute. However, for accurately estimating $\hat{b}$, a more realistic approach would be to find its second moment. The detector is a typical correlation receiver and analysis work of Hernandez *et al.* [109] shows that the probability of error P_e is given by $Q(\sqrt{SNR})$ when the DCT distribution of the image is considered to be a GGD. Here, $Q(\cdot)$ is given by

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} exp\{\frac{-t^2}{2}\}dt. \tag{5.6}$$

The SNR differs for our case due to the full frame DCT implementation. If the select set of AC coefficients represents a set A, then the SNR over set A is defined as

$$SNR_A = \frac{E\left[\sum_{(k,l)\in A} \lambda_s(k,l)\right]}{\sqrt{Var.\left[\sum_{(k,l)\in A} \lambda_n(k,l)\right]}}. \tag{5.7}$$

Now, the SNR_A defined in Eq. (5.7) is higher than the conventional SNR obtained from block-based DCT embedding and therefore the probability of error during the detection of the embedded bit is lower. The increase in SNR_A comes from the reduced standard deviation of the DCT coefficient set in the case of full frame DCT. The advantages and disadvantages of using a full frame DCT for error concealment are described next.

5.2.1 Full Frame DCT vs. Block-based DCT Embedding

There are two distinct advantages of using a full frame DCT over a block-based DCT for embedding: (1) We can embed more number of bits in the AC coefficients of a full frame DCT than in the block-based DCT, and (2) The detector performance is better in the case of full frame DCT than block-based DCT.

We provide here a qualitative analysis for the aforementioned two advantages. The first advantage is evident from the fact that in the full frame DCT case, we do not need to embed a single bit multiple times to get the same level of decoding efficiency. Since we use only 4 coefficients (a 2 × 2 set) for embedding each bit, instead of the central fold of 8 × 8 AC coefficients, we can embed more information in fewer coefficients than the conventional block-based embedding.

The proof of the second advantage is rather tricky. Let us first consider the distribution of AC coefficients in both full frame and block-based DCT cases. The coefficients in the DCT are a sum of cosine-weighted pixel values. We know by the central limit theorem that, the greater the number of terms in the summation, the smaller is the standard deviation of the normalized sum. Since the coefficients in the full frame DCT are a weighted sum of a much larger number of pixel values than those in the block-based DCT, full frame DCT generates coefficients with smaller deviations. The smaller the deviation, larger is the SNR, and therefore, higher is the accuracy of the detection of the incoming bits. Hence intuitively, a full frame DCT should have better detection performance when compared to the block-based DCT.

We next consider the set of AC coefficients we use to embed the watermark data, i.e., coefficients in the set A. This set is located in the mid-frequencies of the full frame DCT. Since the energy level of the DCT coefficients drops from the first to the fourth quadrant, the standard deviations of the mid-frequencies in a full frame DCT are expected to be comparable to the average of the standard deviations of the all the full frame DCT AC coefficients. Though the standard deviations are varying in the set A from the first quadrant to the fourth, the deviation in the sum of coefficients throughout A is much less than the deviation in the sum of AC mid-coefficients of the individual blocks in the block-based DCT. Since the standard deviation is smaller in the sum of coefficients in set A of a full frame DCT, SNR_A in Eq. (5.7) is higher and we expect higher level of efficiency in accurately detecting the embedded image.

One conceivable disadvantage of the full frame DCT when compared to block-based DCT is the computational complexity. For an $N \times N$ size image, the computational complexity of a block-based DCT is of the order of $\mathcal{O}(N^2)$ while that of a full frame DCT is $\mathcal{O}(N \cdot \log(N))$. However, the added complexity in our method arises due to the process of embedding prior to encoding. Therefore, the overall complexity of our system is $\mathcal{O}(N^2) + \mathcal{O}(N \cdot \log(N))$. By observing the order of complexity, we identify that the increment is small compared to the overall complexity of encoding, and therefore could be overlooked considering the gains.

5.2.2 α Variation

Even though the standard deviation is decreasing from the first to the fourth quadrant in the set A, the overall decrease is very little. For this reason, each 2×2 set of coefficients that we use to embed a single bit is assumed to have the same standard deviation.

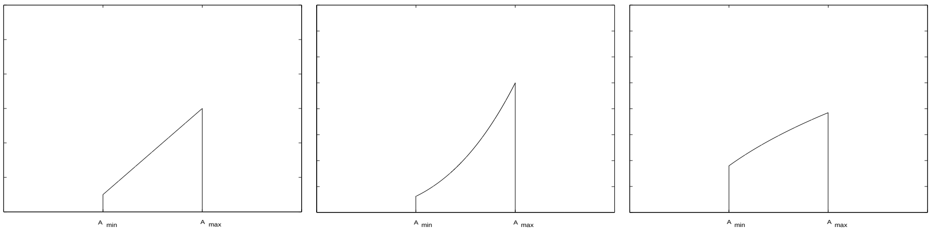


Figure 5.2: 3 variations of α, (a) linear, (b) convex, and (c) concave.

However, we chose the scale factor α such that the ratios of the embedding bits' energy is constant for the AC coefficients in A, i.e., for the 2×2 set of coefficients in the central AC frequencies,

$$\frac{\alpha_i \tilde{w}_i(k,l)}{\alpha_j \tilde{w}_j(k,l)} = \text{constant}, \qquad \forall \;\; i \neq j. \tag{5.8}$$

By choosing so, we make the resulting overall distribution of the AC coefficients in A to have a smaller standard deviation and therefore more accurately detectable. From Eq. (5.8), the variation in α is defined to be an inverse relation to the pixel strength at a particular location.

This implementation of variation in α however requires not only higher computational complexity but also a knowledge of other coefficients' strengths while calculating α for the current coefficient. An alternate way to produce a similar effect for α variation without the computational overhead is to follow one of the 3 variations of α in Fig. 5.2, linear, convex, or concave. α at A_{min} corresponds to the strength of the first AC coefficient in A and α at A_{max} corresponds to the strength of the last AC coefficient in A. The results in all 3 cases are presented in Section 5.4.

5.3 Rate-Distortion Analysis

Now that we have estimated the probability of error in watermark detection and BER incurred during the extraction process, we can predict the effects of the BER on the video signal in terms of the total distortion. The distortion measure we use is the total squared error (TSE) in the video sequence. This estimated distortion is considered while analyzing the rate-distortion variations in the WEC implementation.

Fig. 5.3 shows the rate-distortion curves for the host image. The lower curve in the

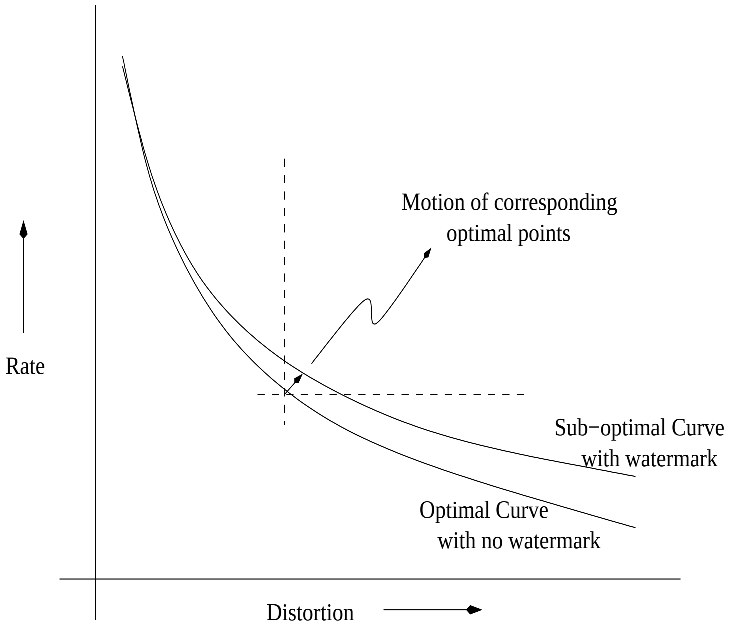

Motion of corresponding optimal points
Rate
Sub−optimal Curve with watermark
Optimal Curve with no watermark
Distortion

Figure 5.3: Rate-Distortion curves for the host image, with and without embedding.

figure represents the rate-distortion variation when no watermark is embedded, while the upper curve shows the variation when a watermark is embedded. At higher bit rates, the watermarked data is invisible since the watermark is inserted with a small value of α, and therefore the distortion that occurs due to the watermarking defects is minimum, thereby pushing the curve closer towards the optimal curve (without embedding). However, as the rate decreases, the value of α needs to be increased for maintaining the high accuracy of the watermark detection process and therefore, watermarking defects creep in, thus increasing the distortion in the host video signal. The figure also traces a sample point movement when the host image is embedded with a watermark.

However, the movement of the optimal point in the rate-distortion curve in the case of the error concealed image is quite different as seen from Fig. 5.4. Let (R_0, D_0) be a sample optimum point in the R-D curve when the watermark is not embedded. Let us also assume for the sake of simplicity that in this case, both WEC and ECSI were applied to the lossy received image and the error concealed image has passed through the same amount of degradations and packet loss percentages. The co-ordinates of the new optimal point after error concealment in the case of WEC is (R_1, D_1) and in the case of ECSI is (R_2, D_2). Note that $D_1 < D_0$ while R_1 is slightly greater than R_0. Also note

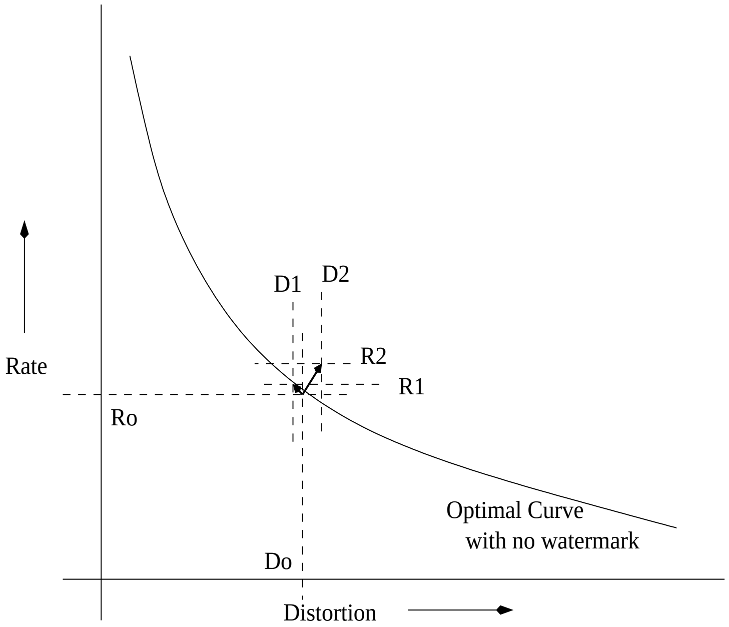


Figure 5.4: Rate-Distortion curves for the error concealed image. The motion of the optimal point is indicated both in cases of WEC and ECSI.

that the WEC optimum point now lies on or just above the same curve as when there is no watermark.

Even though $R_1 > R_0$, the increase is very small and can be considered negligible. In the case of WEC, there is enhancement in the detected watermark performance than when no watermark is embedded and therefore we have lesser distortion, while the increased entropy in the watermark adds to the increased bit rate. Note that this increase in bit rate increases for higher packet loss percentages in the case of informed watermarking since there would be more bit errors in the detected watermark.

In the case of ECSI, the bit rate increases considerably due to addition of side information. We therefore have $R_2 > R_1 > R_0$. Furthermore, the distortion increases too, due to the reasons explained in Section 3.5.3, both in terms of the higher compression and/or packet error prone side information. For these reasons, $D_2 > D_0 > D_1$. Note also that even though the WEC point is not optimum, it is quite close to the optimum and lies on or just above the optimum curve. A further comparison of these two techniques in the case of losses is presented after analyzing the performance of WEC in case of high packet loss probabilities.

Fig. 5.5 shows the performance of the WEC algorithm implementation in various

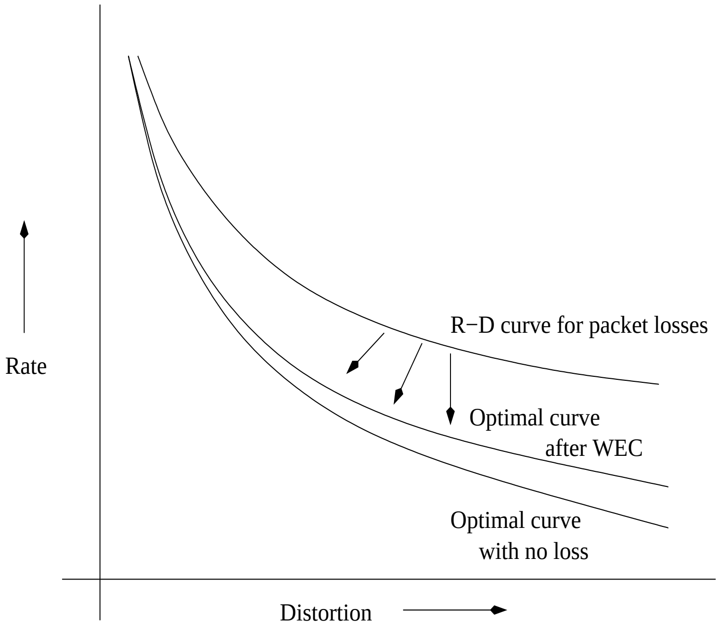


Figure 5.5: Rate-Distortion performance of WEC algorithm.

distortion scenarios. The plot includes three curves, all for the same video signal and the same WEC technique. The bottom curve represents the optimal curve in case of no packet losses with watermark embedding. The top curve shows the R-D behavior with packet losses. It can be seen that the distortion increases faster than the decrease in bit rate in the case of packet losses. The middle curve shows the R-D variation with WEC implemented on the packet loss affected video signal. Note that WEC pushes the packet loss affected curve closer to the optimum (when there is no loss) in the way suggested by the arrows. Also note that WEC operated on top of any error correcting codes (ECC) and/or ECSI would display the same type of characteristics, i.e., it tends to push the R-D behavior closer to the optimal error free performance.

Referring back to the comparison with ECSI, let R_0 be the rate at which the encoded raw video is transmitted and R_1 be the rate of the auxiliary information. Then the total transmission bit rate for ECSI would be $R_2 = R_0 + R_1$. In the case of WEC, the transmission bit rate is R_0. Fig. 5.6 shows the comparison of the R-D behavior for both WEC and ECSI. For any $x\%$ packet errors, both WEC and ECSI tend to push the R-D curve of the lossy signal closer to the optimum, but WEC does a better job. This is again due to the two reasons explained before. In this case, consider just the transfer of

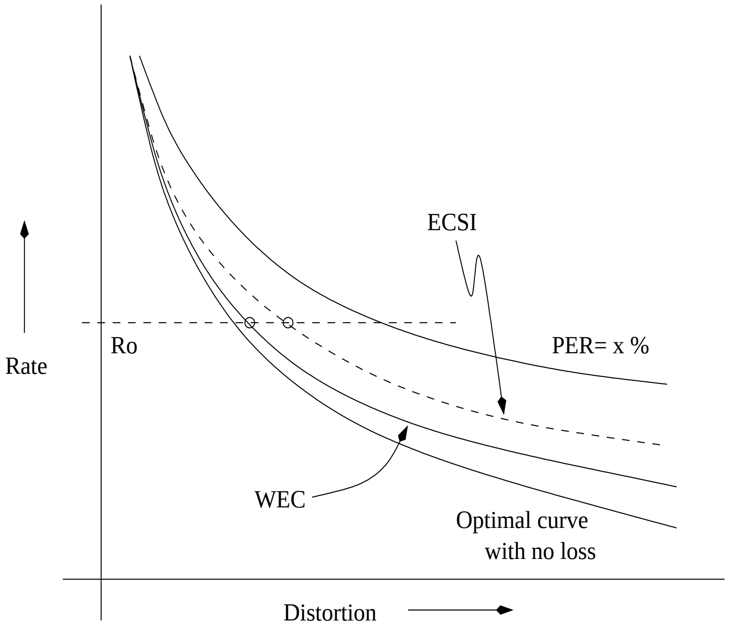


Figure 5.6: Rate-Distortion performance comparison of WEC and ECSI.

the raw source encoded data with a rate R_0. From the figure, we see that ECSI produces higher distortion than WEC for the same rate R_0, while any ECC applied to ECSI would put the curve between the current ECSI curve and the WEC curve.

Consider now, the comparison of WEC and the most recently proposed and incorporated Reed-Soloman (n, k) block codes (r - s codes) for ECC (with or without ECSI implemented on top of ECC). Fig. 5.7 shows the R-D behavior of such codes for removing the bit errors that occur during the packet losses. It shows the top curve to be for a fixed $x\%$ BER. The relation between the packet error rate (PER) and BER for a typical wireless error prone channel (that can be modelled as a Gilbert-Elliot two-state Markov chain) is given by:

$$PER = 1 - (1 - BER)^L \tag{5.9}$$

where L is the number of bits in the packet. Let R_1 be the encoded source rate and R_2 be the rate after adding either ECC or auxiliary (side) information. For rate R_1, the bit rate at which WEC code is transmitted, we observe that the distortion produced is much less than D_1, the distortion produced by ECC. However, since the transmission rate in case of ECC is R_2, the effective R-D optimal point for ECC would for a distortion of D_1 would lie farther away from the distortion point at R_2, as shown in the figure.

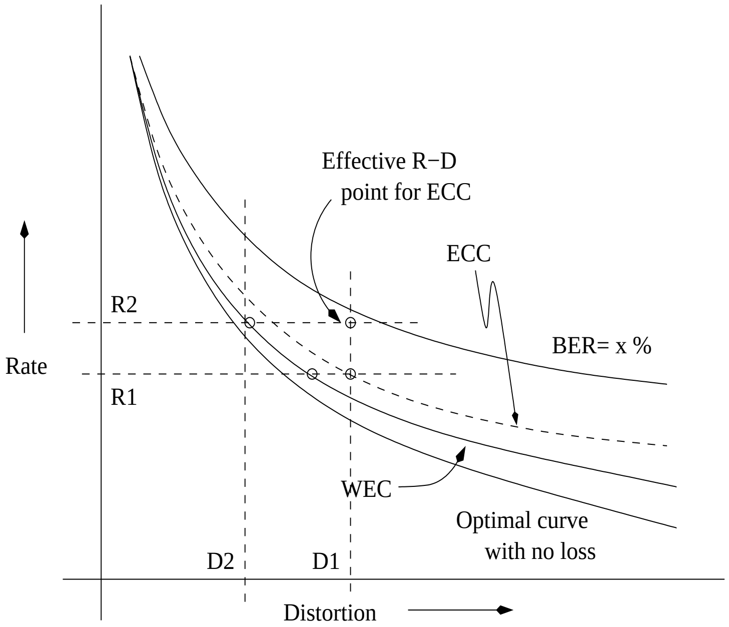
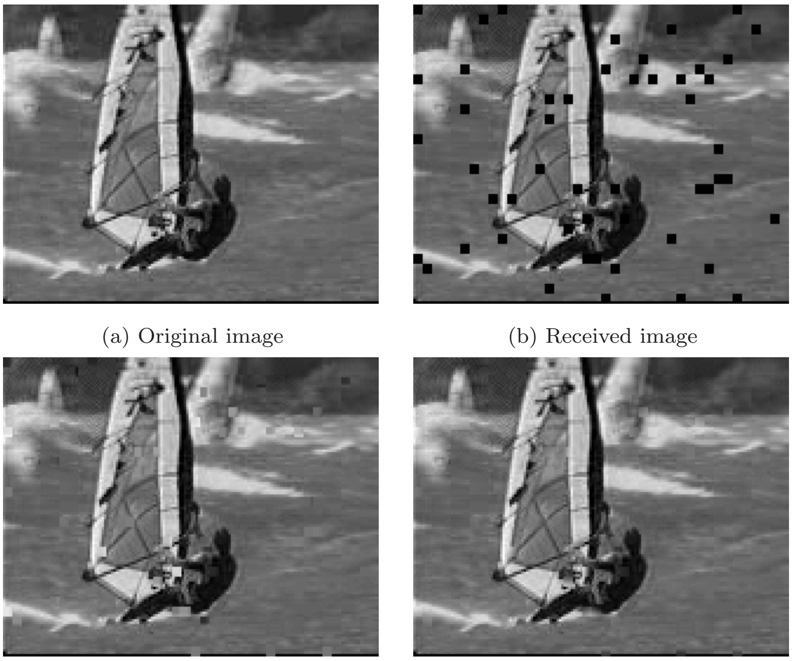


Figure 5.7: Rate-Distortion performance comparison of WEC and error correction codes.

Note that the effective R-D point for ECC is higher in both bit rate and distortion compared to the effective optimal R-D point in case of WEC (this point is the intersection of WEC curve and R_1 in the figure). We therefore conclude that WEC results in a more optimal point in the bit rate-complexity-quality (distortion) triangle.

5.4 Simulations

A sample set of images of fixed size, 240 × 320, is used for the simulation and the watermark is inserted in the central AC frequencies of the full frame DCT. For the error concealment application, low-frequency embedding would cause visible artifacts in the image, while high-frequency embedding would make it more prone to channel induced defects. The *ns*-2 simulator is used to generate packet losses with a two-state Gilbert-Elliot Gaussian packet loss model with predefined mean and variance. The packet size was fixed at 1500 kB for a given image transmission.



(a) Original image

(b) Received image

(c) Block-based DCT embedding

(d) Full frame DCT embedding

Figure 5.8: The Original image is shown in (a). Received image obtained for a mean packet loss probability of 0.12 and variance 2%. The PSNR values of the images are: (b) 19.6012, (c) 28.8226, and (d) 32.3122, respectively. The value of α used in both cases was 3.

5.4.1 Full-frame and block-based DCT

Fig. 5.8 shows the performance of WEC on the image *Sail*. A block size of 8×8 was used for block-based DCT embedding and α was fixed at 3 for both cases. Fig. 5.8(b) shows the lossy received image, while Figs. 5.8(c) and (d) show the error concealed images for block-based DCT embedding and full frame DCT embedding, respectively. For a loss probability of 0.12, we observe about 3.5 dB increase in PSNR in the full frame DCT case.

Fig. 5.9 shows the PSNR vs. loss rate curves for the received image, block-based DCT embedding, and full frame DCT embedding. In the case of block-based embedding, the value of α was fixed at 3, while in the case of full frame DCT embedding, the value of α was varied according to the three cases in Fig. 5.2. As seen from the figure, we obtain better performance when α is concave, especially at higher loss probabilities. The performance of WEC is very similar in cases when α is both linear and convex. The energy in the AC coefficients in set A for *Sail* image may be decreasing convexly, which

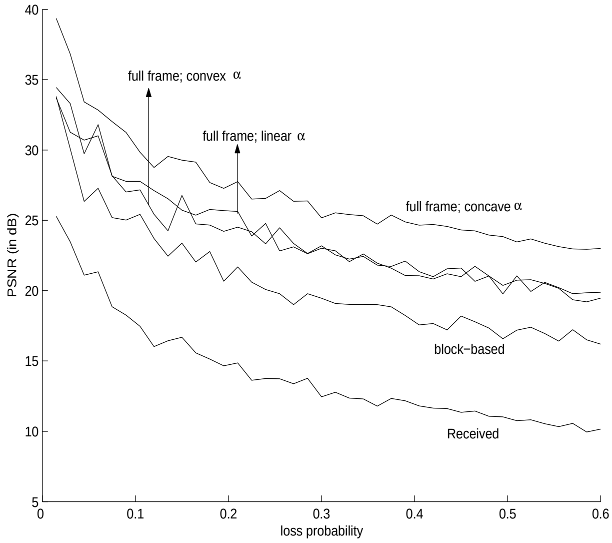


Figure 5.9: PSNR vs. loss probability curves for WEC with block-based DCT and full frame DCT. α is varied according to Fig. 5.2 in the case of full frame DCT.

might lead to this performance.

From the figures, we can observe that the full frame DCT embedding outperforms block-based DCT embedding for error concealment applications.

5.5 Summary

We have provided a mathematical analysis of the detector performance for an WEC system, which uses a full frame DCT for embedding a low resolution version of itself. Consequently, we present the advantages and disadvantages of the full frame DCT against the conventional block-based DCT embedding. For the application of error concealment, we conclude that the full frame DCT embedding is more advantageous using both analysis and simulations. We can not only embed higher number of bits in a full frame DCT but also detect them more accurately. Additionally, we present three variations of watermark strengths to reduce the probability of error during detection.

The rate-distortion model for the proposed WEC algorithms is analyzed based on (1) the BER in the watermark detection process along with the probability of error calculation, and (2) the packet losses incurred during video transmission. The distortion created in a video from these two aspects is estimated and it is shown that WEC algorithms not only push the R-D performance of the lossy received video towards the optimum lossless curve, but also perform better that ECSI and other ECC methods in achieving a more optimal curve. An added advantage of the WEC algorithms is that they could be used on top of any/all ECC and ECSI algorithms, thereby improving their performances.

Chapter 6

Subjective Video Quality Evaluation

In Chapter 4, we developed two ways of spatial embedding watermark data, namely, embedding intra-coded frame watermark reference in itself and embedding a prediction frame watermark reference in the current intra-coded frame. We have shown that even though the average luminance PSNR values of the error concealed videos in both these cases are similar, there is a considerable difference in the luminance PSNR values of the frames over time, i.e., the luminance PSNR values of individual frames in the video varied over frames in the case of the former embedding scheme, while in the case of the latter, the luminance PSNR variation remained almost constant with increasing frame numbers [114]. To further substantiate the notion that the subjects consider a relatively constant quality video over time to be more preferable than a varying quality one, and to prove the fact that the latter embedding scheme receives higher subjective quality rating than the former, we have carried out a psychophysical evaluation of the quality of the reconstructed video.

In this chapter, we present in detail, the psychophysical experiment that we have conducted to test the performance of the two different WEC implementations in varying packet loss scenarios. The motivation that drives the need for this experiment is the fact that the two WEC techniques produce error concealed videos that have the same or similar objective quality measures even though subjectively, they seem to have varying qualities.[1] The aim of this experiment is therefore: *To identify, subjectively assess,*

[1] Another motivation in performing a psychophysical experiment is the fact that PSNR is not a good measure for objective quality. Though there have been attempts to develop an objective quality metric that performs as well as the subjective testing [115]-[118], nothing concrete has been developed so far.

quantify, and compare the performance of two different WEC techniques for varying video contents, codecs, and packet loss scenarios. The chapter also explains in detail, the criteria for the selection of the videos with specific contents, the process of generating the test sequences, the experimental apparatus, test set-up, testing procedure, and the analysis of the data obtained from the experiment.

6.1 Psychophysical Evaluation

The scope of this psychophysical experiment has been to evaluate the increment in perceived quality of the video sequences that are error concealed using the WEC techniques proposed in Chapter 3 and 4. To this end, the goals of the experiment have been the following:

- The subjective evaluation of the increment in the perceptual quality that the proposed technique provides over conventional error concealment in low bit rate video codecs,

- The comparison between the perceptual quality of intra-coded reference embedding in the intra-coded frame and inter-coded reference embedding in the intra-coded frame, and

- The verification of codec-independency during implementation of the proposed algorithm (two low bit rate codecs, MPEG-4 and H.264 have been used for this test).

The errors incurred are due to packet losses during the transmission of video sequences. The WEC technique used is the one proposed in Section 3.6. The implementation to the MPEG-4 and H.264 videos is similar to the one described in Sections 4.2.1 and 4.2.2. We are primarily interested in evaluating the effects and variations in the perceived quality of the low bit rate compressed video sequences with packet loss variations, implementation methodology, and codec dependency with varying content. Packet losses are generated at random, and therefore, the way they affect the block compressed encoded video is different in each case. This section describes the source videos used, the generation of video sequences with the required variations, and the subjective testing procedure in detail.

6.1.1 Video Sequences

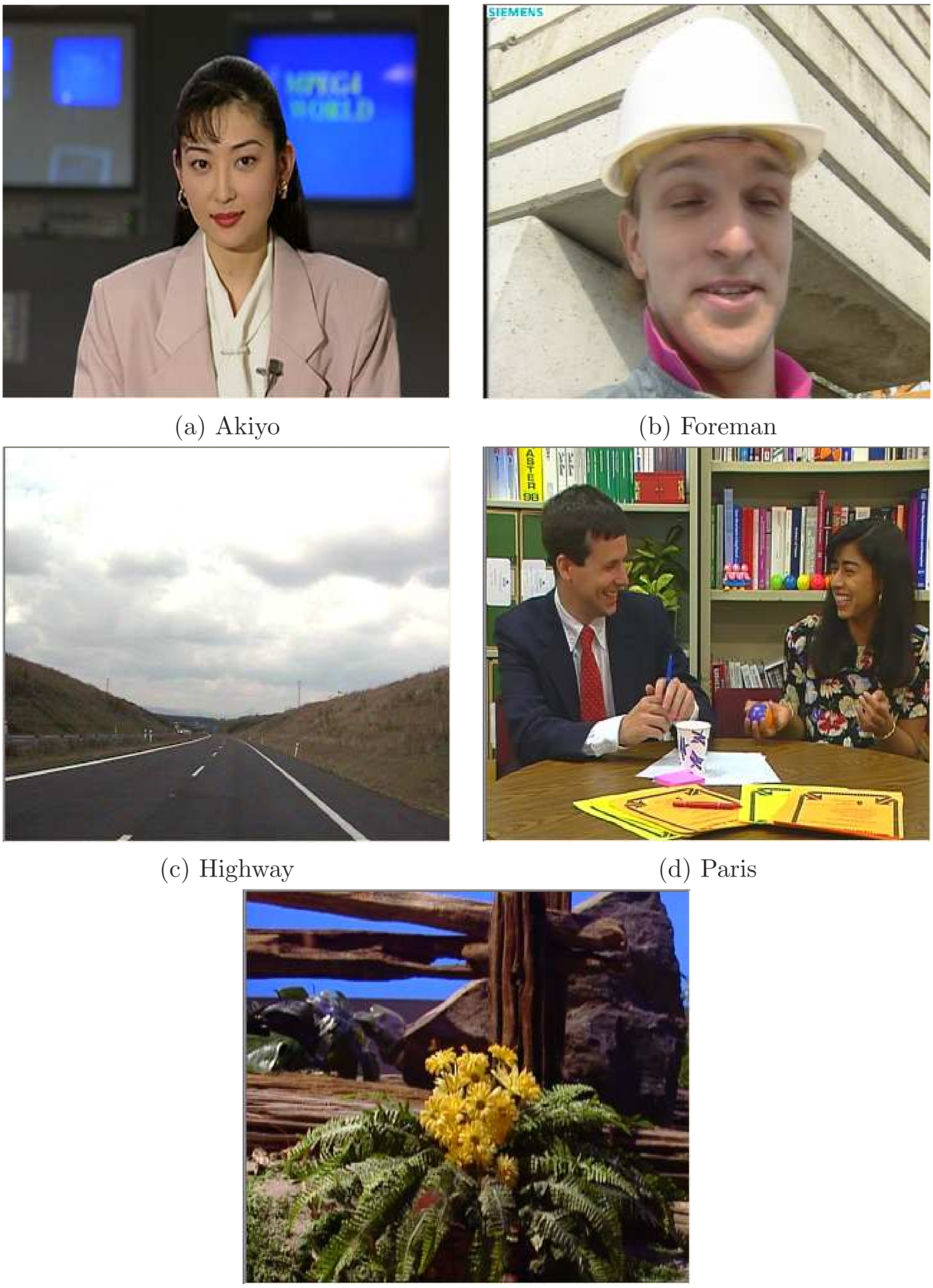
Five original video sequences have been used in the experiment. The selected videos had varying content (smooth as well as highly textured areas) and motion (slow, fast, linear, non-linear, and zoom). They are compressed image format (CIF) sequences, each with a resolution of 288 × 352 pixels, i.e., each frame of the video sequence has 288 lines and 352 pixels in each line. All five videos were 10 seconds long, and played at 24 fps.

The video sequences considered for the experiment are shown in Fig. 6.1. Fig. 6.1(a) shows the video *Akiyo*. The video presents a lady reading news in a television broadcast. It is considered for relative low frequency content, low motion, and segmented bright color attributes. Fig. 6.1(b) shows the *Foreman* video sequence. This video presents a foreman in a construction site explaining the details of the construction. It is selected for its facial features and fast motion. Fig. 6.1(c) shows the *Highway.* The video shows the front view of the highway while seated in a car moving at a relatively constant speed. It is considered for its smooth areas and relatively constant linear motion. Fig. 6.1(d) shows the video *Paris*. This video presents two people talking in an office/library environment. This video is selected based on its high frequency content, prominent non-linear motion, and color attributes. The fifth video sequence, *Tempete*, is shown in Fig. 6.1(e). This is a video shot of a small flowery plant in a ranch on a windy day. It is considered for its texture features, non-linear motion, and zooming camera motion.

The key variations we considered for generating the video test sequences required for the experiment are as follows:

- *Packet losses:* Five different packet loss variations, 0%, 0.1%, 0.2%, 0.3%, and 0.6%, were considered for the experiment. The % loss represents the ratio of the number of packets lost to the total number of transmitted packets. Note that the first case of 0% loss corresponds to the original video transmission. Note also that these loss percentages were created to be approximately equal. The criterion for matching the quality loss however, is the logarithm of the total squared error (log TSE), as is explained in Sections 4.2.1 and 6.1.2 [116], [115].

- *WEC implementations:* Three variations to the WEC algorithm implementations are used in this experiment. The first variation is without any WEC implemented,



(a) Akiyo (b) Foreman (c) Highway (d) Paris (e) Tempete

Figure 6.1: The original video sequences used for the experiment. The first frame of each video are presented here as snapshots of the sequences.

i.e., there is no embedded data and the concealment is performed by the pixel interpolation and other advanced techniques that are in-built in the codec, either MPEG-4 or H.264. The second variation is with the intra-coded frame embedded inside itself, i.e., the WEC algorithm explained in Section 4.2.1. The third technique is the predicted frame embedded in the intra-coded frame as explained in Section 4.2.2.

- *Codecs:* Two different low bit rate video codecs are used here, MPEG-4 and H.264. The MPEG-4 codec we used is developed by the joint motion experts group (JMEP) and is the freely available latest version *TM*-5.0 group implemented by the Microsoft Corp (MFC). The H.264 codec is jointly developed by the international telecommunications union (ITU) group and the international standards organization, motion pictures expert group (ISO/MPEG) and is also the freely available latest version *JM*-7.4 reference software.

- *Compression:* Even though there is an effect of compression on the performance of the proposed WEC algorithm as discussed in Chapter 3, due to the shortage of allotted experiment time and the number of sequences in view, this parameter was kept fixed. All the video test sequences were compressed to a bit rate of 120 kbps using their respective codecs. Note that this is a highly compressed very low bit rate for the CIF video sequences. However, we use this high compression to illustrate the improvement in performance of our algorithm over the conventional ones even at very low bit rates.

- *Resolution and Frame rates:* The resolution of the video test sequences is selected to be CIF, 288×352. Quadrature CIF (QCIF) was another consideration for resolution, but we decided to go against it as the subjects would find it difficult to identify any sort of watermarking defects at such small resolutions. The frame rate at which the video test sequences are played was at 24 fps. We chose this frame rate as this is the average frame rate suggested in literature for low bit rate compressed videos [94], [103], [119].

- *Time and Memory requirements:* Based on the ITU-T standards for subjective testing experiments, any evaluation of packet loss or transmission defects requires a video sequence to be played for a minimum of 8 seconds before asking the subject to rate it. We therefore consider 10 second long videos. The standards also suggest that the concentration time by the subjects would not last more than one hour. So, to get meaningful data out of the experiment, we chose to play the videos for a period of 45 minutes. Considering a response time for each video by a subject to be $8 - 9$ seconds would give us approximately 150 test sequences. Each sequence is stored and presented in .avi format, which takes around 36 MB for each video sequence or approximately 9.6 GB for the entire experiment. However, since the videos are stored and played from an NEC server (PC), the loading and display times are negligible.

With these considerations, the total number of video test sequences required for the psychophysical experiment are calculated to be:

$$5 \text{ Originals} \times 2 \text{ Codecs} \times 3 \text{ WEC impl.} \times 5 \text{ Packet losses} = \mathbf{150} \text{ Test sequences} \tag{6.1}$$

Apart from these 150 test sequences, a set of 5 test sequences with varying packet losses (selected from the above 150 sequences) were shown to the subjects as examples of types of defects that they might observe in the actual videos. Furthermore, the subjects were also shown a set of 5 test sequences (again, selected from the 150 sequences) as part of a practice/training session.

6.1.2 Generation of Videos

The process that was used to generate the test sequences is described in this section. Fig. 6.2 shows the entire process in sequential order. A *"*"* in the figure represents all the different combinations, for example, *"O*"* represents all the 5 originals, *"w*"* represents all the 3 WEC variations, and *"l*"* represents all the 5 different loss variations. The *"_m_"* stands for MPEG-4 encoding and *"_h_"* for H.264 encoding.

The five originals, *A: Akiyo, F: Foreman, H: Highway, P: Paris,* and *T: Tempete* are passed through the watermark embedding process, which uses three different WEC variations, no embedding (*"_wne"*), embedding in the I-frame the watermark generated from current I-frame (*"_wii"*), and embedding in the current frame the watermark generated from the subsequent third frame (*"_wpi"*). The informed WEC scheme is used for embedding and the watermark is generated using the DPCM encoded reference. The output of the embedding process would yield 15 sequences (five in case of each WEC variations, corresponding to the five originals). These 15 sequences are sent to both MPEG-4 and H.264 encoders and are compressed to 120 kbps to give a total of 30 streams (15 for each encoder *"_m_"* and *"_h_"*). Note that since we are comparing the results of the WEC algorithm with the error concealment in the standard codecs, we turn on all the necessary rate control and error concealment parameters in the configuration (*.cfg*) and parameter (*.par*) files of the encoders.

These 30 streams are passed through the simulated random packet loss process to give 150 streams. Each of the 15 streams in case of MPEG-4 and H.264 go through the loss process independently. Gaussian random packet drop algorithm with a predefined mean loss and loss variance is used to generate the five different packet loss variations. For each packet transmitted, the packet drop algorithm generates a Gaussian random variable from a distribution with a fixed mean and variance and if the value of this variable is outside a specified range, the packet is dropped. The probability that a packet will be dropped is varied by changing the range, called the limit. Five packet loss probabilities are used for each of the 30 videos to give 150 streams. The five losses are denoted by *"l0"* through *"l4"* and the lossy streams are *"O*_m_w*_l*"* (75 lossy streams for MPEG-4) and *"O*_h_w*_l*"* (75 lossy streams for H.264).

Tables 6.1 - 6.5 show the variation of the range (or the limit values) for the mean and the loss variance of the distribution to generate the packet loss percentages of 0%, 0.1%, 0.2%, 0.3%, and 0.6%. However, due to the random nature of the loss generation, the losses might occur either in the I-frames or the predicted and the bi-directional prediction frames. If the losses occur in the Intra-coded frames, then there is substantial error propagation and therefore, the overall error occurring in the video is higher than when losses occur only in the P- and B-frames. This is true even when I-frames are affected with lower packet loss percentages than when P- and B-frames are affected with higher packet loss percentages. Therefore, the yard stick for evaluating the quality ratings obtained from the subjects is validated to be a measure of the total squared error (TSE) in the video rather than the packet loss percentages. This is more relevant to the data

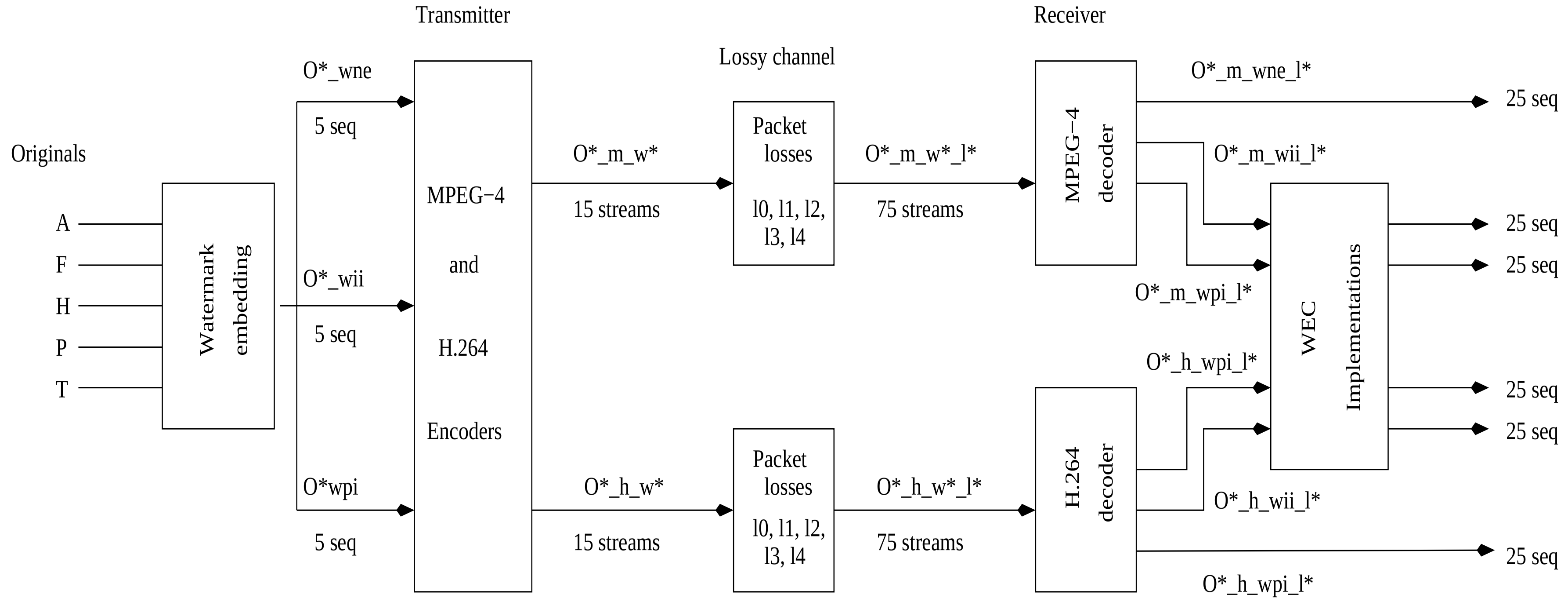


Figure 6.2: Block diagram of the procedure for generating the video test sequences for the psychophysical experiment.

Table 6.1: The loss variance, the range (limits), and the packet loss percentages created for the *Akiyo* original.

Akiyo		
Variance	Limit	Loss%
0.00	0.00	0.00
11.00	12.00	0.09674
12.00	11.50	0.19348
11.00	11.75	0.29022
11.00	11.50	0.58044

Table 6.2: The loss variance, the range (limits), and the packet loss percentages created for the *Foreman* original.

Foreman		
Variance	Limit	Loss%
0.00	0.00	0.00
11.00	13.00	0.09659
11.00	12.50	0.19318
11.00	12.00	0.28977
12.00	11.75	0.57954

analysis of the subjects' data and therefore are discussed more in detail in Section 6.2.1.

The lossy streams are then decoded using the corresponding decoders (75 each for MPEG-4 and H.264). The sequences with no watermark embedded are directly considered as test sequences after decoding. These are 50 sequences in all that do not undergo any WEC implementation (that only have baseline error concealment implementations of the codecs), 25 sequences corresponding to MPEG-4 (“O*_m_wne_l*”) and 25 corresponding to H.264 (“O*_h_wne_l*”). The other 100 sequences pass through the two different WEC implementations, I-frame reference watermark embedded in the I-frame, and the subsequent P-frame reference watermark embedded in the current I-frame, 50 sequences corresponding to each of these variations. For each of the 50 sequences, 25 correspond to MPEG-4 and the other 25 correspond to H.264. For the 50 sequences in each case of WEC implementation, the watermark is extracted and used for error concealment. These error concealed images are used as test sequences, 25 each for MPEG-4 and H.264, in the cases of *“_wii”* and *“_wpi”*, thus producing the remaining 100 sequences. The total of 150 sequences are used as test sequences for the experiment.

Table 6.3: The loss variance, the range (limits), and the packet loss percentages created for the *Highway* original.

Highway		
Variance	Limit	Loss%
0.00	0.00	0.00
12.00	12.50	0.09643
11.00	12.00	0.19286
11.00	11.50	0.28930
12.00	11.50	0.57859

Table 6.4: The loss variance, the range (limits), and the packet loss percentages created for the *Paris* original.

Paris		
Variance	Limit	Loss%
0.00	0.00	0.00
11.00	12.00	0.09678
11.00	11.50	0.19355
12.00	12.50	0.29033
13.00	11.75	0.58066

Table 6.5: The loss variance, the range (limits), and the packet loss percentages created for the *Tempete* original.

Tempete		
Variance	Limit	Loss%
0.00	0.00	0.00
11.00	12.00	0.09674
11.00	12.50	0.19348
11.00	11.00	0.29022
12.00	11.50	0.58044

6.1.3 Testing Procedure

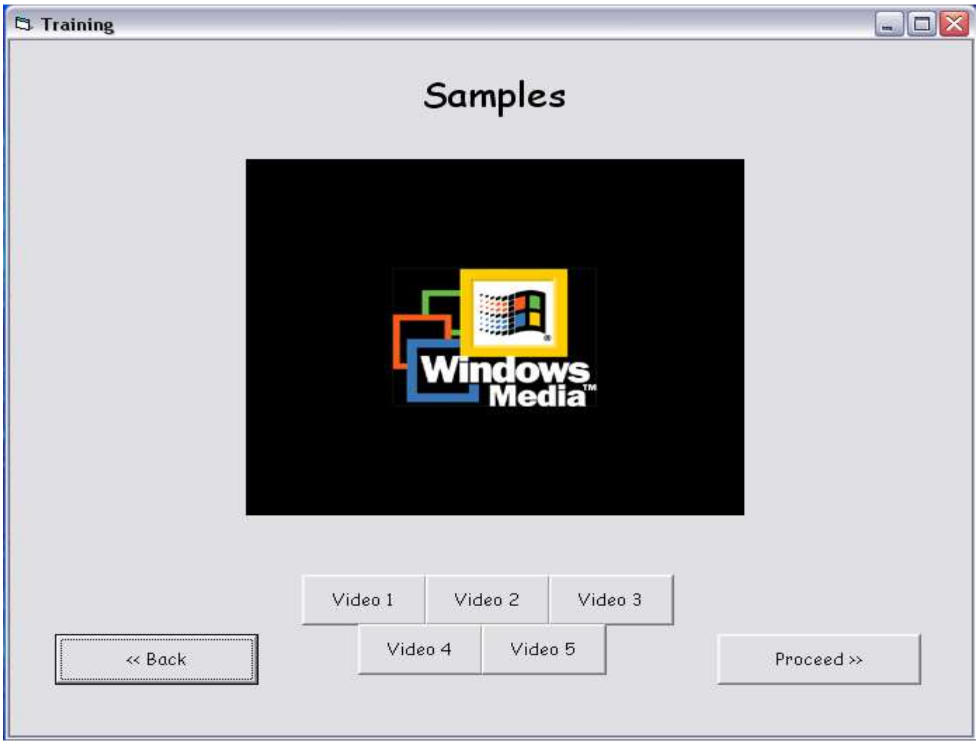
The test video sequences were displayed on a 17" DELL LCD monitor. The native resolution of the monitor was 1024 × 768. The monitor was characterized by determining the chromaticities of the three colors. It was determined to have a γ value of 1.5 by a photometer (the allowable range of γ for best viewing is $1.3 - 2.5$). The details of the measurement and theoretical aspects behind γ calibration are described in Appendix A.

The video sequences were stored in an NEC file server (PC) and displayed using a dedicated interface developed for the purpose of this experiment. The server and the monitor were placed in a dimly illuminated room that approximated the moderate viewing conditions. The monitor was placed on a table that stood approximately three feet from ground. The subjects sat in a stable chair (non-swivelling) placed at a distance of 30 inches from the monitor, which is approximately 6 times the height of the displayed video (as per the ITU-T standards).

The software interface for the experiment had a front end that incorporates the media player to present the CIF videos in their actual sizes (288 lines of 352 pixels each). Fig. 6.3 shows the display of the program for one of the reference screens.

From the 150 test sequences that were generated, a set of 150 test trials were formed using 2-set combinations of these test sequences. The variations that were described in the previous subsections were used to differentiate between the trials, i.e., the three WEC variations (no-embedding, WEC with I-frame embedding, and P-frame embedding) were displayed as three comparisons ($wne - wii$, $wii - wpi$, $wpi - wne$). For each trial, all the other variations were kept constant, i.e., for each trial, the left and the right test sequences were processed using the same original, the same codec, and the same packet loss percentage. We therefore had 50 sets of three video sequences. In each set there were three possible pairs, so there were 150 test trials.

Each test trial had a set of two test sequences displayed side-by-side and played simultaneously. The sequences were synchronous, i.e., they started and ended at the same time. The test trials were created based on the dual-stimuli presentation standard according to the ITU-T subjective testing standards. However, even though the subjects were asked to rate each of the test sequences in each trial, the values they gave represented their preference, i.e., the video sequence (either left or right) that received a higher quality



Training

Samples

Figure 6.3: Display of the front-end of the program for one of the sample screens.

number indicated that the subject rated it to have higher quality than the other. The motivation for performing this process is obtained from the results that we observed from a similar dual-stimuli experiment that conducted earlier.

The experiment was divided into 4 stages: Originals, Samples, Practice, and Main. In the “Originals” stage, the five originals without embedding were displayed. The subjects were asked to assign a value of 100 to the best of the five videos. They were then told that this value of 100 would act as reference in judging the quality of the test sequences in the main experiment. They were instructed to rate the quality of the video sequences relative to this value of 100, i.e., if the quality of the video sequence that they saw was half as good as the quality of the best of the originals, they would give a value of 50, or if the video that they saw was twice as good, they would assign a number 200 to it.

In the “Samples” stage, 5 video sequences were shown that had varying quality. These video sequences were chosen to be a sample set of the test sequences. The set consisted of the five originals as well as all 5 packet loss percentages implemented on the videos and processed with the three WEC variations in both the codecs. These sample sequences were shown to the subjects to give them an idea of the range of qualities of the video test sequences that they are going to see throughout the main experiment. This way, the subjects would be able to assess the range of quality values that they would be assigning to the video sequences.

The "Practice" session was very similar to the main experiment. In this stage, five test trials were shown. The test trials were again chosen to be a sample set of the video sequences in the main experiment. These were presented in order for the subjects to be completely understand and familiarize themselves with the video dual-display system and what their task would be in the main experiment. It also removed/reduced any learning errors that would be incurred in the first 4-5 video sequences in a typical subjective experiment.

In the main experiment, all the 150 test trials were presented in random order. For each trial, all subjects viewed the same video sequence with the same packet loss errors. After viewing the two video clips in the test trials, all subjects were asked the same question: "On a scale relative to 100 (that you set with originals and samples), rate the quality of both the left and right videos in each trial, giving a higher value to the one that you feel has a higher quality." A sample screen of the main experiment is shown Fig. 6.4. They were instructed to rate the quality of the video sequences relative to this value of 100, i.e., if the quality of the video sequence that they saw was half as good as the quality of the best of the originals, they would give a value of 50, or if the video that they saw was twice as good, they would assign a number 200 to it.

After the main experiment, the subjects were interviewed with regard to the type of defects they observed in the video sequences that they saw and how annoying were there defects.

6.1.4 Subjects

A group of 39 subjects drawn from a pool of students in the introductory course in Psychology Department at the University of California, Santa Barbara (UCSB) were recruited for this experiment. They were considered naïve of most kinds of digital video defects and the associated terminology for the purpose of subjective quality evaluation of the video sequences. They were asked to wear glasses or contact lenses if they need

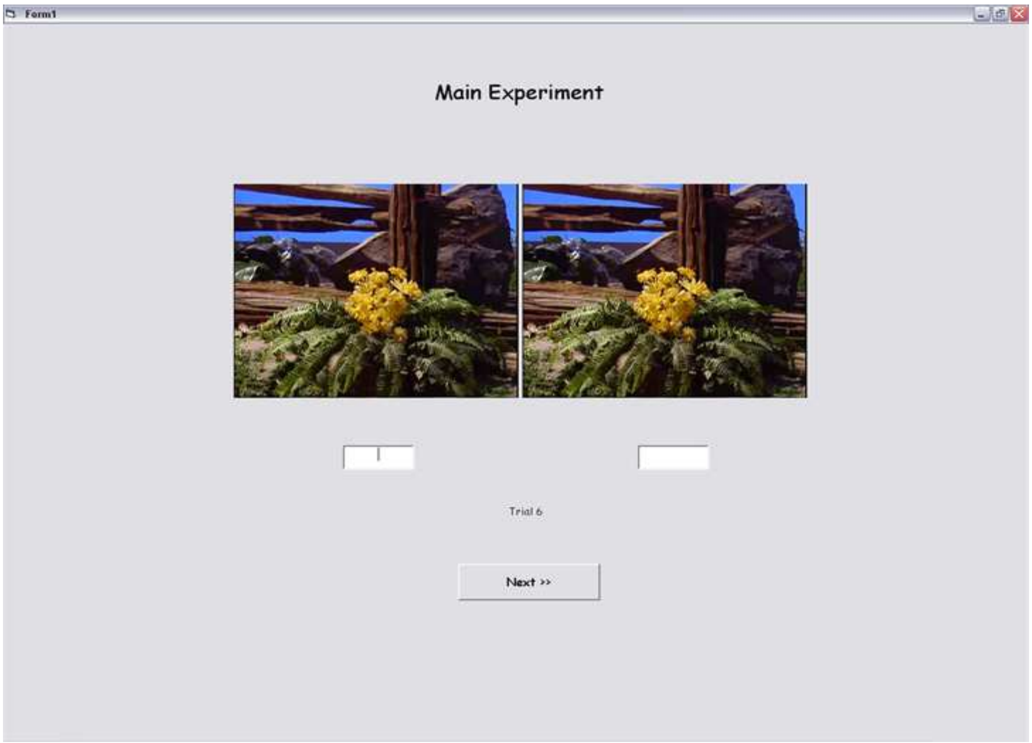


Figure 6.4: Display of the screen for the main experiment (full-screen).

them to watch TV or computer monitor. Of the set of 39, 30 subjects were instructed to evaluate the quality of the video test sequences relative to a value of 100, while the other 9 were considered for evaluating the quality of the video test sequences in between 0 and 100, i.e., they were asked to pick a value for the quality of the video from a scale that spans from 0 to 100, with 100 being the best quality and 0, the worst. However, the displayed test video sequences, the examples, the practice/training sessions, and the interview questions were the same for all subjects.

6.2 Results and Analysis

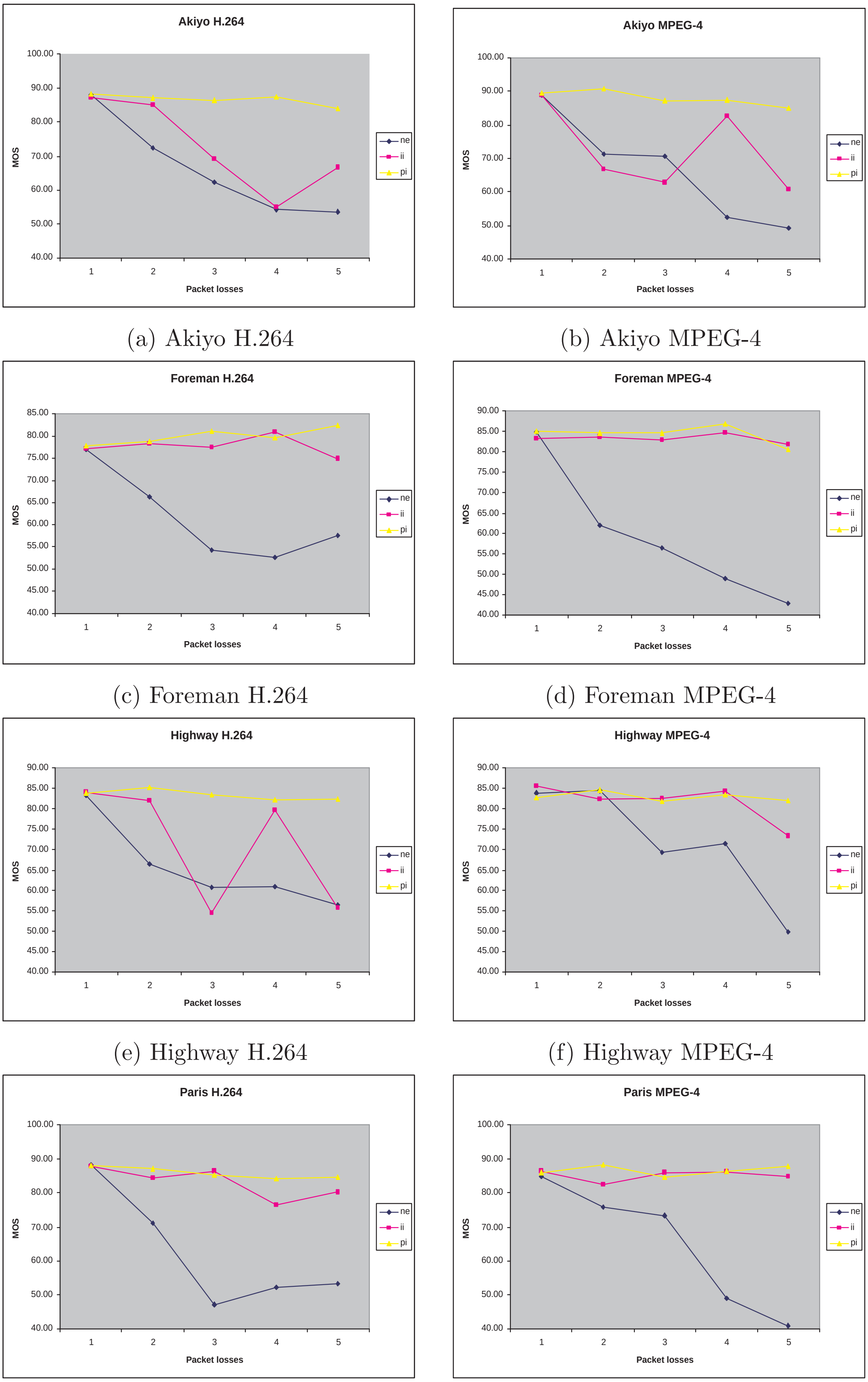
The results of the experiment are analyzed and presented in this section. The average of the subjects' scores for each video, called the mean opinion score (MOS), is considered for rating the quality of the video sequences. The analysis, however, is divided into different sections based on the dependency of the scores on the type of video content, the codec used, the packet loss percentages, and the WEC algorithm used. This is done for obtaining a better understanding of the relation between MOS and its individual dependencies.

6.2.1 WEC performance

Fig. 6.5 shows the MOS scores obtained for each video compressed using both H.264 (on the left) and MPEG-4 (on the right). The three WEC variations are shown together and compared in each plot. The *"ne"* in the plots represents the case where no WEC was implemented, the *"ii"* represents the case where WEC with I-frame reference embedding is implemented, and the *"pi"* represents the case where WEC with P-frame reference embedding is implemented. The x-axis represents the packet loss percentage levels, 1 corresponding to 0% up to 5 corresponding to 0.6%.

From Fig. 6.5, we observe that the WEC with the P-frame in the I-frame embedding gave the best overall MOS results for almost all packet loss percentages. In almost all plots, WEC implementation perceptually enhanced the quality of the video. Note also that WEC with the P-frame in the I-frame embedding was evaluated to give more constant quality (in terms of MOS values) over varying packet loss percentages. Even though WEC with the I-frame in the I-frame embedding gives a comparable performance to WEC with the P-frame in the I-frame embedding, it did not achieve the constant quality attribute for varying packet loss percentages.

Fig. 6.5 clearly shows the improvement in performance with WEC over the conventional codecs for varying packet loss percentages along with the constant perceptual quality attribute of WEC with the P-frame in the I-frame embedding. However, the comparison of the MOS values obtained for the WEC algorithm implementations is not the only way to evaluate the quality of the video as a function of the packet loss percentage. As can be seen from the plot, in some occasions, as the packet loss percentages increase, the MOS values increase. This phenomenon is contrary to the normal expectation that MOS values decrease as the packet losses increase. The reason for this is the random packet drop as explained in Section 6.1.2, and therefore, the indirectly dependent variable we use to predict the quality loss is the error measure of log TSE, which is directly related to the amount of error that occurred in the video due to packet drops.



(a) Akiyo H.264

(b) Akiyo MPEG-4

(c) Foreman H.264

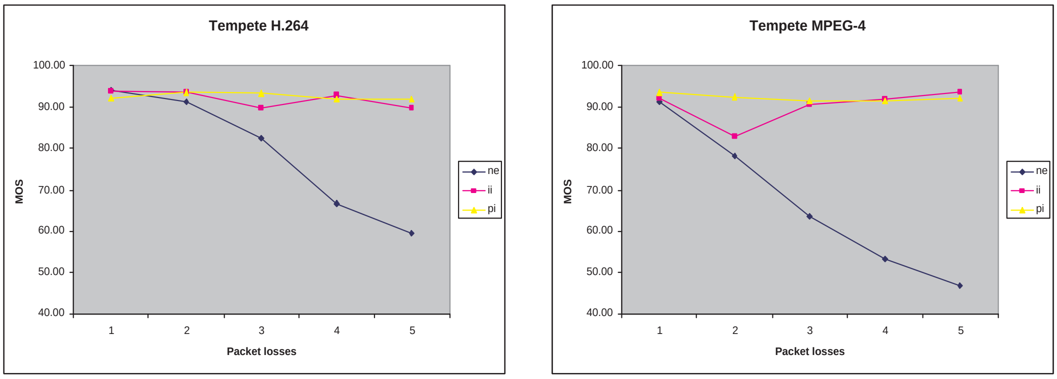
(d) Foreman MPEG-4

(e) Highway H.264

(f) Highway MPEG-4

(g) Paris H.264

(h) Paris MPEG-4



(i) Tempete H.264

(j) Tempete MPEG-4

Figure 6.5: The variation of MOS values with increasing packet loss percentages. Here 1 through 5 on x-axis represents packet loss of 0% through 0.6%.

Log TSE is a measure of the total error in the video sequence and is given by:

$$\log\text{TSE} = \log_{10}(\text{TSE}_1 + \text{TSE}_2 + \text{TSE}_3) \tag{6.2}$$

where TSE_i is defined as

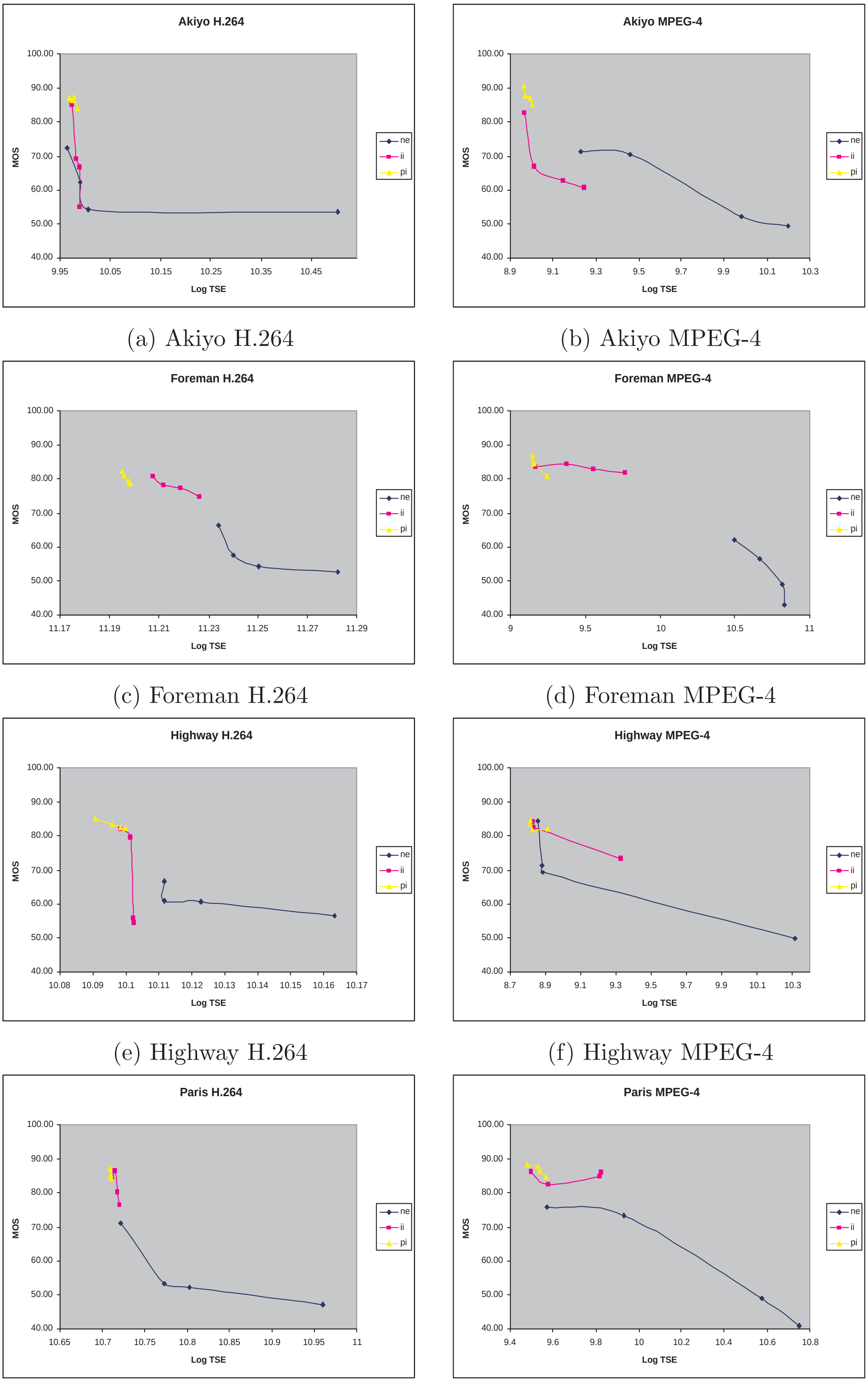
$$\text{TSE}_1 = \sum_{i,j}(f_{Y,p} - f_{Y,o})^2 \quad \text{TSE}_2 = \sum_{i,j}(f_{U,p} - f_{U,o})^2 \quad \text{TSE}_3 = \sum_{i,j}(f_{V,p} - f_{V,o})^2 \tag{6.3}$$

Here, Y, U, and V refer to the luminance and the chrominance components of the video, the subscript o for the frame f represents the original, the subscript p represents the processed frame, and i, j represent the pixel coordinates. From the equation, we see that the higher the difference from the original, the higher the value of log TSE. Due to the random drop of packets that affect the frame in unequal amounts both spatially (when packet loss affects the high frequency content of a single frame) or temporally (when packet loss occurs in P- and B-frames as opposed to I-frames), the amount of error incurred in the video is a more appropriate measure of the transmission losses rather than the percentage of packets dropped. Log TSE captures the amount of loss incurred by the packet drop algorithm and therefore is used here as a quality matching criterion.

Fig. 6.6 shows the plots of the MOS scores obtained for the videos against log TSE of the video. We observe that the curves for no WEC implementation are towards to the bottom right corner of the plots, which means that they have high error in them and therefore are low on MOS values of the quality. The curves for the WEC implementation with the I-frame reference embedding in the I-frame are in between (and closer to the third set of WEC with the P-frame reference embedded in the I-frame), which means that the error in them has been reduced and therefore received higher MOS values when compared to no WEC. The third set, as we expected, has the least amount of error and received the highest MOS values, and therefore are towards the top left corner of the plot. We also observe that in some cases, when the error in the two cases of WEC processed videos is the same, the P-frame embedding in the I-frame WEC implementation was evaluated to have a higher perceptual quality.

A closer look at the plots also shows that while the MOS values for no WEC implementation were spread out for varying packet loss percentages, the P-frame embedding in the I-frame WEC implementation is clustered towards the top left corner of the plot. This occurs due to the reasons explained in Sections 3.5 and 4.2.1 where the variances in the video quality due to packet losses are described. The significance of this effect is that the WEC implementation with the P-frame reference embedding in the I-frame performs well even in case of higher losses thereby producing a similar quality for the error concealed video at all losses in the range measured. This phenomenon therefore reiterates the constant quality attribute of the WEC implementation with the P-frame in the I-frame embedding. Though not as prominent as the WEC implementation with the P-frame in the I-frame embedding, a similar effect can be seen also in the case of the WEC implementation with the I-frame in the I-frame embedding. Note that this effect is evident when WEC is applied to both the codecs, but the average error is about the same.

We found that in this dual stimulus experiment, the MOS values obtained for a single video differ when its neighbor is different. The MOS values that we tabulated and plotted so far were the average of the different samples. We now look at the differences in the MOS values of each video when it is compared to different sequences in different trials. Fig. 6.7 shows the differences in MOS values of each video in the cases of H.264 (a) and MPEG-4 (b). The *x*-axis represents videos with the packet loss percentages, i.e., *f2* in Fig. 6.7 *x*-axis represents *Foreman* video with loss of 0.2%. The legend shows a set of three WEC algorithms for each video and loss percentage, *ne-i*, *i-p*, and *p-ne*. The first



(a) Akiyo H.264

(b) Akiyo MPEG-4

(c) Foreman H.264

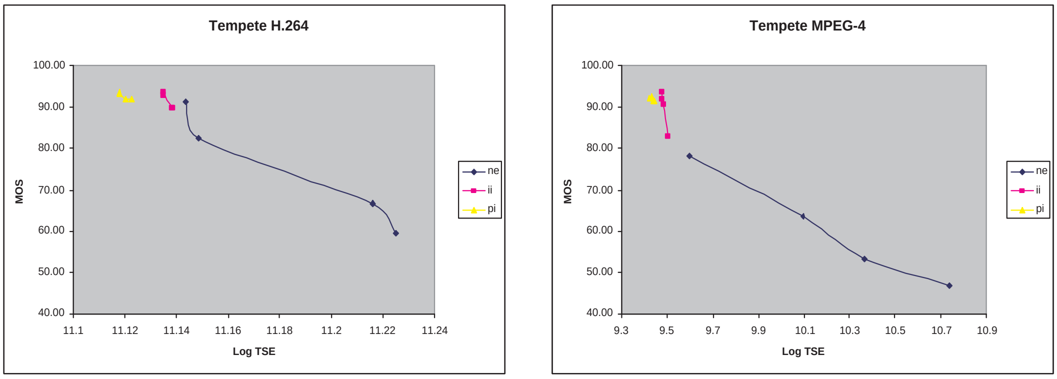
(d) Foreman MPEG-4

(e) Highway H.264

(f) Highway MPEG-4

(g) Paris H.264

(h) Paris MPEG-4



(i) Tempete H.264

(j) Tempete MPEG-4

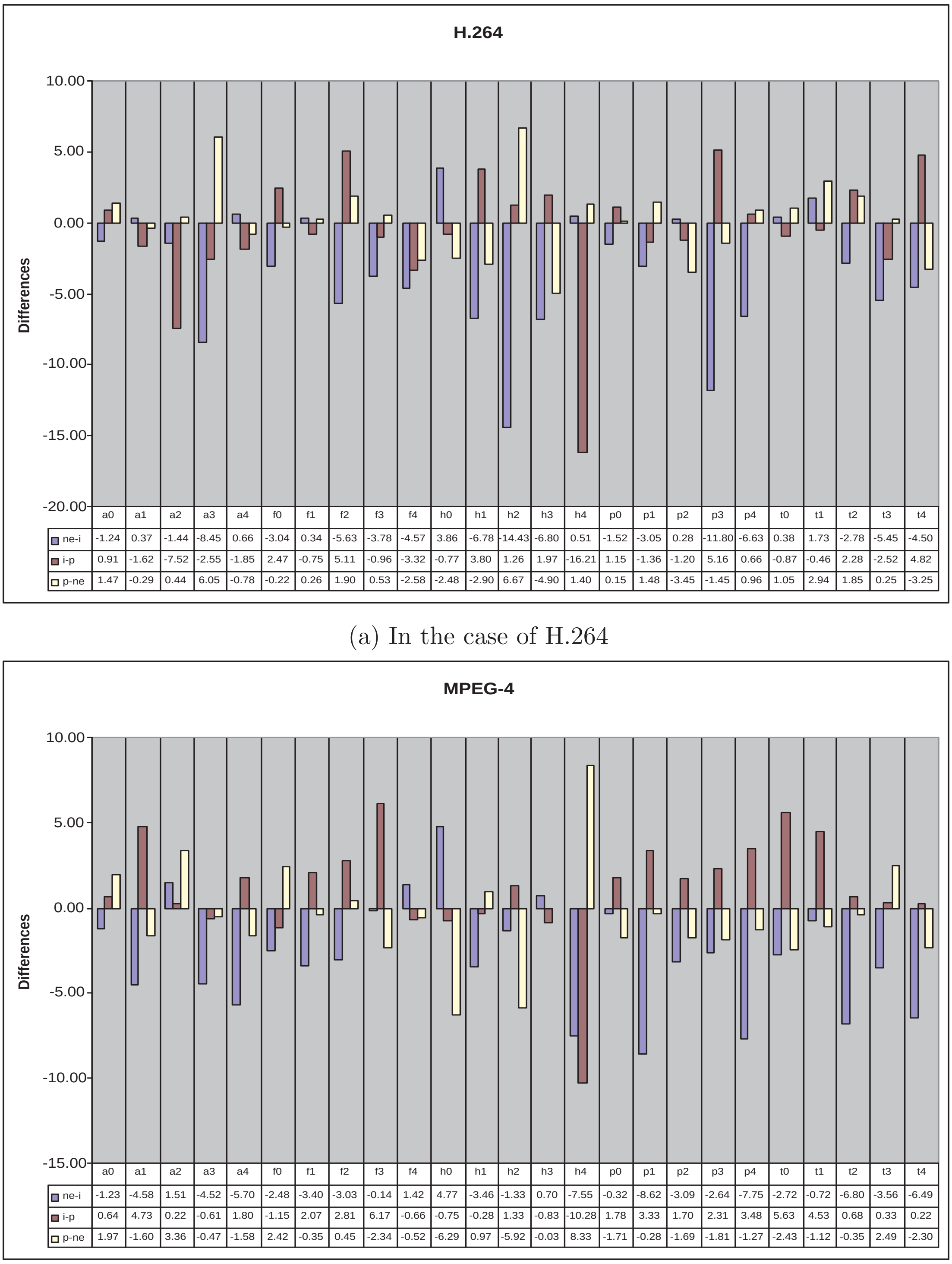
Figure 6.6: The variation of MOS values with increasing Log TSE values. The Log TSE values are evaluated to be the physical measure of the error in the video.

character (or set of characters in case of *"ne"*) in each of these algorithms represents the video that has its MOS difference tabulated. The second character(s) represents the neighbor of the video represented by the first character(s) during display, that acts as the first term in the MOS difference. This is better explained with an example as follows.

Consider the value of -1.44 in the *a2* column for the *ne-i* WEC algorithm row in Fig. 6.7(a). This value indicates the difference in the MOS values obtained for *Akiyo* video with a packet loss percentage of 0.2% with no WEC algorithm implementation (*"ne"*) when it is compared with *"i"* and when it is compared with *"p"*. The second character *"i"* states that the MOS value of *akiyo_hne_l2* when it is compared to *"p"* algorithm implementation, *akiyo_hpi_l2*, is subtracted from the MOS value of *akiyo_hne_l2* when it is compared to *"i"* algorithm implementation, *akiyo_hii_l2*. In other words,

$$MOS(Akiyo\_hne\_l2_{(a2\_hne,a2\_hii)}) - MOS(Akiyo\_hne\_l2_{(a2\_hne,a2\_hpi)}) = -1.44. \quad (6.4)$$

We observe from the plots in both H.264 and MPEG-4 cases that the MOS difference values are sometimes significantly high. This shows that, in dual stimuli scenario, the quality value obtained for a video depends significantly on the content of the video shown beside it. It is also interesting to observe that the differences for the videos with no WEC algorithm implementation have more bars on the negative for both codecs, which implies that the subjects rated the no WEC videos to have a higher quality when these videos were compared to video with the P-frame in the I-frame WEC algorithm



	a0	a1	a2	a3	a4	f0	f1	f2	f3	f4	h0	h1	h2	h3	h4	p0	p1	p2	p3	p4	t0	t1	t2	t3	t4
ne-i	-1.24	0.37	-1.44	-8.45	0.66	-3.04	0.34	-5.63	-3.78	-4.57	3.86	-6.78	-14.43	-6.80	0.51	-1.52	-3.05	0.28	-11.80	-6.63	0.38	1.73	-2.78	-5.45	-4.50
i-p	0.91	-1.62	-7.52	-2.55	-1.85	2.47	-0.75	5.11	-0.96	-3.32	-0.77	3.80	1.26	1.97	-16.21	1.15	-1.36	-1.20	5.16	0.66	-0.87	-0.46	2.28	-2.52	4.82
p-ne	1.47	-0.29	0.44	6.05	-0.78	-0.22	0.26	1.90	0.53	-2.58	-2.48	-2.90	6.67	-4.90	1.40	0.15	1.48	-3.45	-1.45	0.96	1.05	2.94	1.85	0.25	-3.25

(a) In the case of H.264

	a0	a1	a2	a3	a4	f0	f1	f2	f3	f4	h0	h1	h2	h3	h4	p0	p1	p2	p3	p4	t0	t1	t2	t3	t4
ne-i	-1.23	-4.58	1.51	-4.52	-5.70	-2.48	-3.40	-3.03	-0.14	1.42	4.77	-3.46	-1.33	0.70	-7.55	-0.32	-8.62	-3.09	-2.64	-7.75	-2.72	-0.72	-6.80	-3.56	-6.49
i-p	0.64	4.73	0.22	-0.61	1.80	-1.15	2.07	2.81	6.17	-0.66	-0.75	-0.28	1.33	-0.83	-10.28	1.78	3.33	1.70	2.31	3.48	5.63	4.53	0.68	0.33	0.22
p-ne	1.97	-1.60	3.36	-0.47	-1.58	2.42	-0.35	0.45	-2.34	-0.52	-6.29	0.97	-5.92	-0.03	8.33	-1.71	-0.28	-1.69	-1.81	-1.27	-2.43	-1.12	-0.35	2.49	-2.30

(b) In the case of MPEG-4

Figure 6.7: The differences in the MOS values when measured in comparison between the set of 3 WEC algorithms, *ne-i*, *i-p*, and *p-ne*. The x-axis represents the set of videos, a = akiyo, f = foreman, h = highway, p = paris, and t = tempete, while the number beside it represents the packet loss level. The difference in each comparison is taken for each set of video and packet loss.

implementation. This effect is more apparent in the *Highway* and *Paris* videos for the H.264 codec, and in the *Paris* and *Tempete* videos for the MPEG-4 codec. Moving the video from the left position to the right for the latter comparison could be one of the reasons behind this observation. However, this does not imply that the I-frame WEC algorithm implementation fared better than the P-frame WEC algorithm.

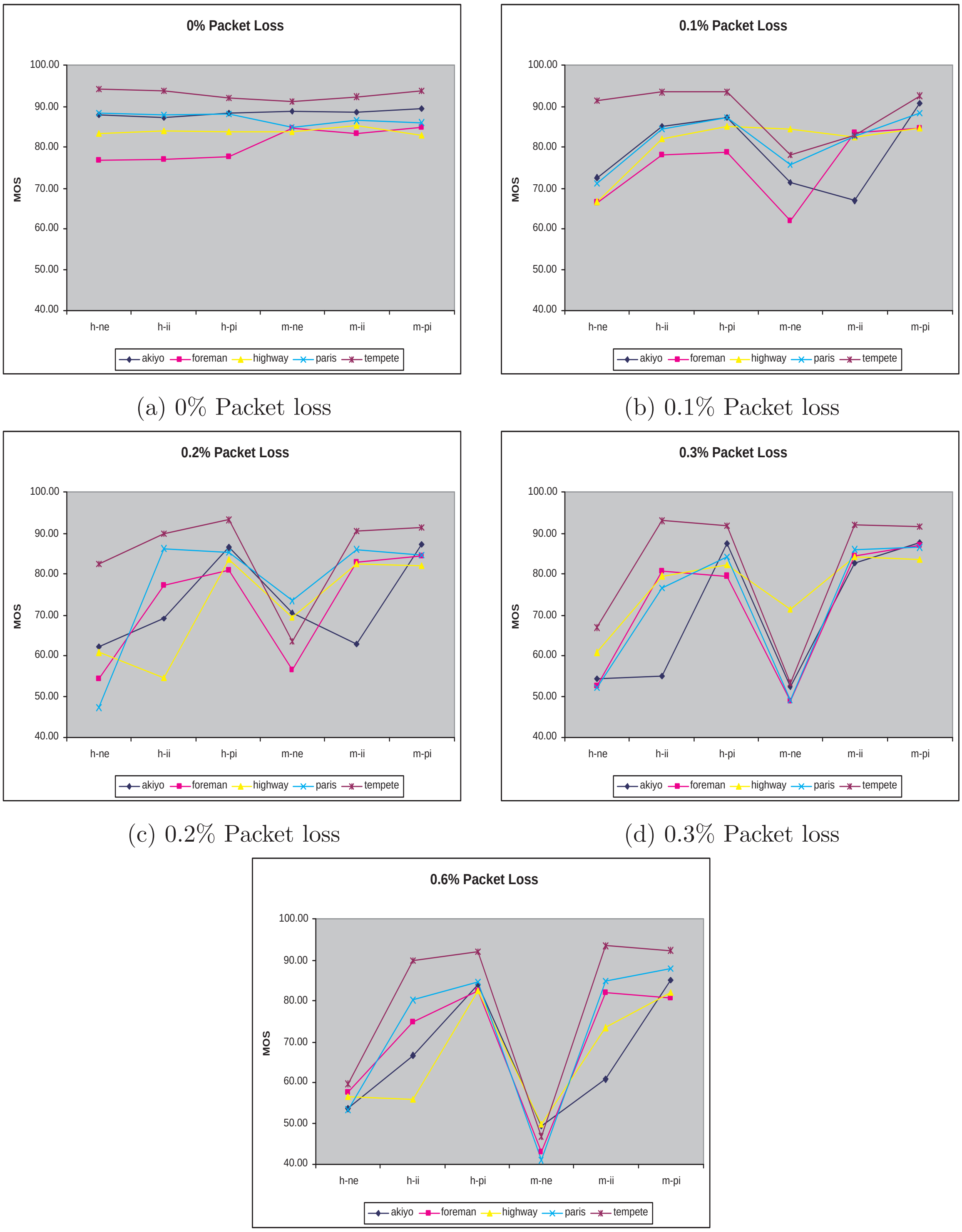
Note that as expected, especially in the case of MPEG-4, while the *i-p* MOS differences are mostly positive, the *p-ne* MOS differences are mostly negative. An extreme case in all the scenarios is the *Highway* video which gives the highest and the least values. The reason for this might be the content of the video (explained in Section 6.2.4) and how packet losses affected this content.

6.2.2 Packet Loss Variation

The values of MOS decrease consistently with increasing packet loss percentages. This can be seen from Fig. 6.5. This phenomenon is apparent in the cases of both H.264 and MPEG-4 implementations. In Fig. 6.6 however, the effect is not as well seen since the plots show variation with log TSE, which might vary based on how and where the packet losses occurred in the videos. For the sake of making the effects of packet losses more visible, we separate the MOS scores for the videos by each packet loss percentage.

Fig. 6.8 shows the variation of the MOS scores for each packet loss percentage variation. The 5 packet loss variations, 0% through 0.6%, are represented here. The x-axis represents all three WEC algorithm implementations for both the codecs. The 5 curves in each plot represent the 5 different videos. We observe that while in Fig. 6.8(a), i.e., for 0%, almost all the videos have relatively constant MOS values for both codec implementations, the improvement in performance due to WEC algorithm implementation becomes increasingly evident as the percentage of packet loss increases. We also notice that as we move along the x-axis, there is an improvement in the MOS scores, i.e., $MOS(no\_WEC) < MOS(I\_WEC) < MOS(P\_WEC)$ in almost all the cases.

Interestingly, the plots in Fig. 6.8 exhibit a dip in the center of the curves and this dip increases as the packet loss percentage increases. This dip exists due to the fact the MOS values for both the MPEG-4 and H.264 are represented in the same plot. Since the order of columns in the x-axis has been from no WEC algorithm implementation to the P-frame



(a) 0% Packet loss

(b) 0.1% Packet loss

(c) 0.2% Packet loss

(d) 0.3% Packet loss

(e) 0.6% Packet loss

Figure 6.8: The MOS values for the three WEC implementations for each video in case of H.264 and MPEG-4. The *"h"* and *"m"* on the x-label represent H.264 and MPEG-4 respectively, with *"ne"*, *"ii"*, and *"pi"* representing the three WEC variations.

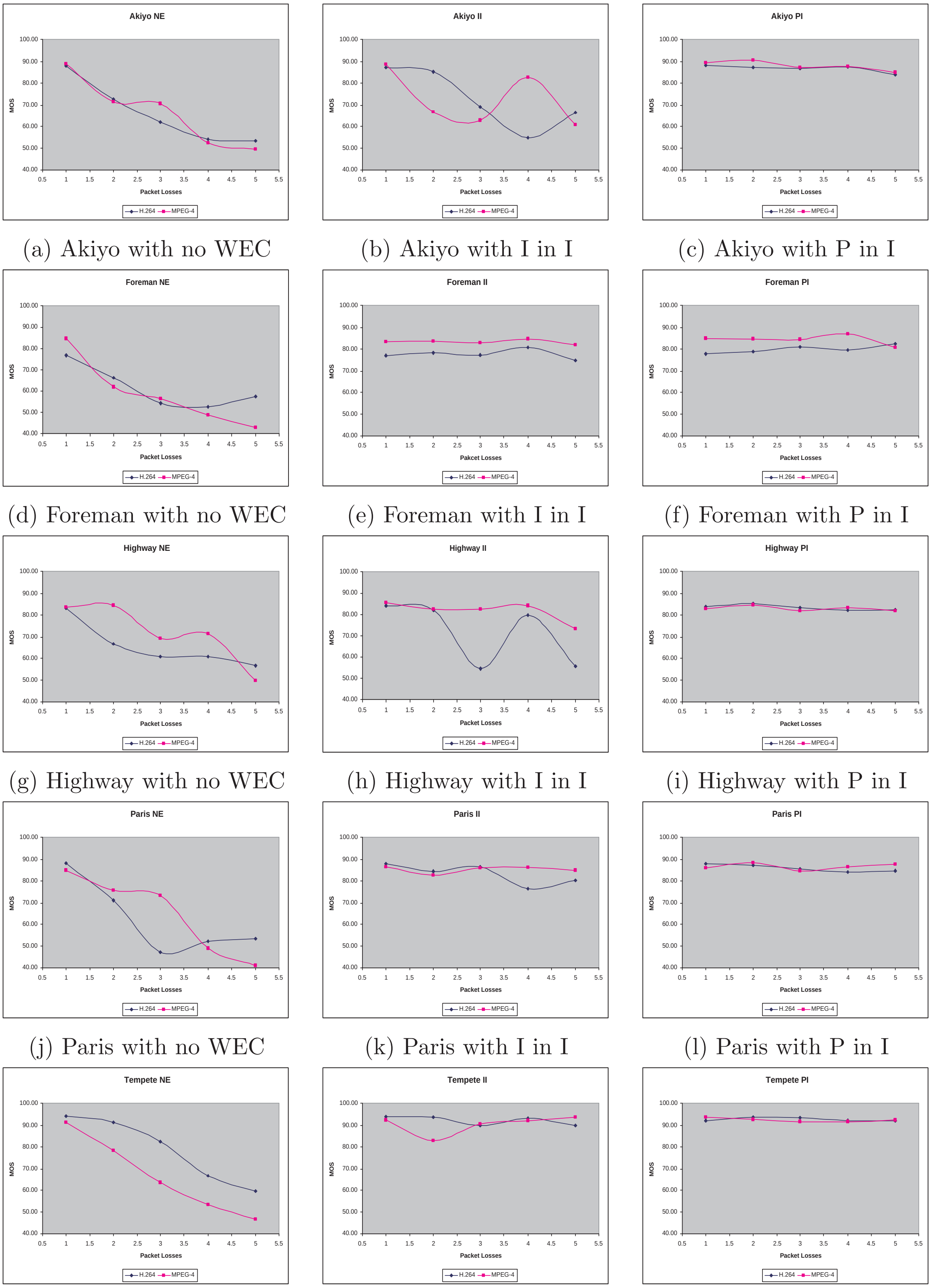
WEC algorithm implementation, we see the drop in performance from the P-frame WEC implementation of H.264 to no WEC implementation in MPEG-4 in the form of this dip. The increase in the dip suggests that the improvement in the performance of the P-frame WEC algorithm implementation increases as the packet loss percentage increases. This again leads us back to the fact that P-frame WEC algorithm implementation gives us almost a constant quality regardless of the amount of packet loss percentages, as seen in Figs. 6.5 and 6.6.

6.2.3 Codec Comparison

The reason behind implementing two codecs for the experiment is to verify the codec-independency of the proposed WEC algorithm implementations. From the Figs. 6.5, 6.6, and 6.8, it is evident that both the variations give a significant enhancement in performance in both H.264 and MPEG-4 codecs. As a further comparison between the two codecs, we can plot the performance of both the codecs in terms of the MOS values in the same plot. Fig. 6.9 shows the variation of the MOS values with and without WEC and packet losses for both the codecs.

Figs. 6.9(a), (d), (g), (j), and (m), on the left represent the plots for all the videos with no WEC algorithm implementation. Figs. 6.9(b), (e), (h), (k), and (n), in the center represent the plots for the videos with the I-frame WEC algorithm, and figs. 6.9(c), (f), (i), (l), and (o), on the right represent the plots for videos with the P-frame WEC algorithm implementation. Note that in the figures of the left, there is a substantial drop in the MOS values with an increase in the packet loss percentages, while in the figures on the right, the MOS values are relatively constant. This substantiates the fact the P-frame WEC algorithm implementation tends to give a constant perceptual quality. For the figures in the center, it is constant of 3 of the 5 videos, but for *Akiyo* and *Highway*, the MOS values varied significantly. This may be partly due to the content of the video and partly due to the way the packet losses affected these videos.

When comparing the codecs, Fig. 6.9 conceals the actual codec error control performance. Except in a few cases ((e) and (f) where it shows that MPEG-4 gives higher MOS values than H.264, and (m) where it shows vice versa), most of the plots show similar performance (trends) for both H.264 and MPEG-4. This however is not true if



(a) Akiyo with no WEC (b) Akiyo with I in I (c) Akiyo with P in I

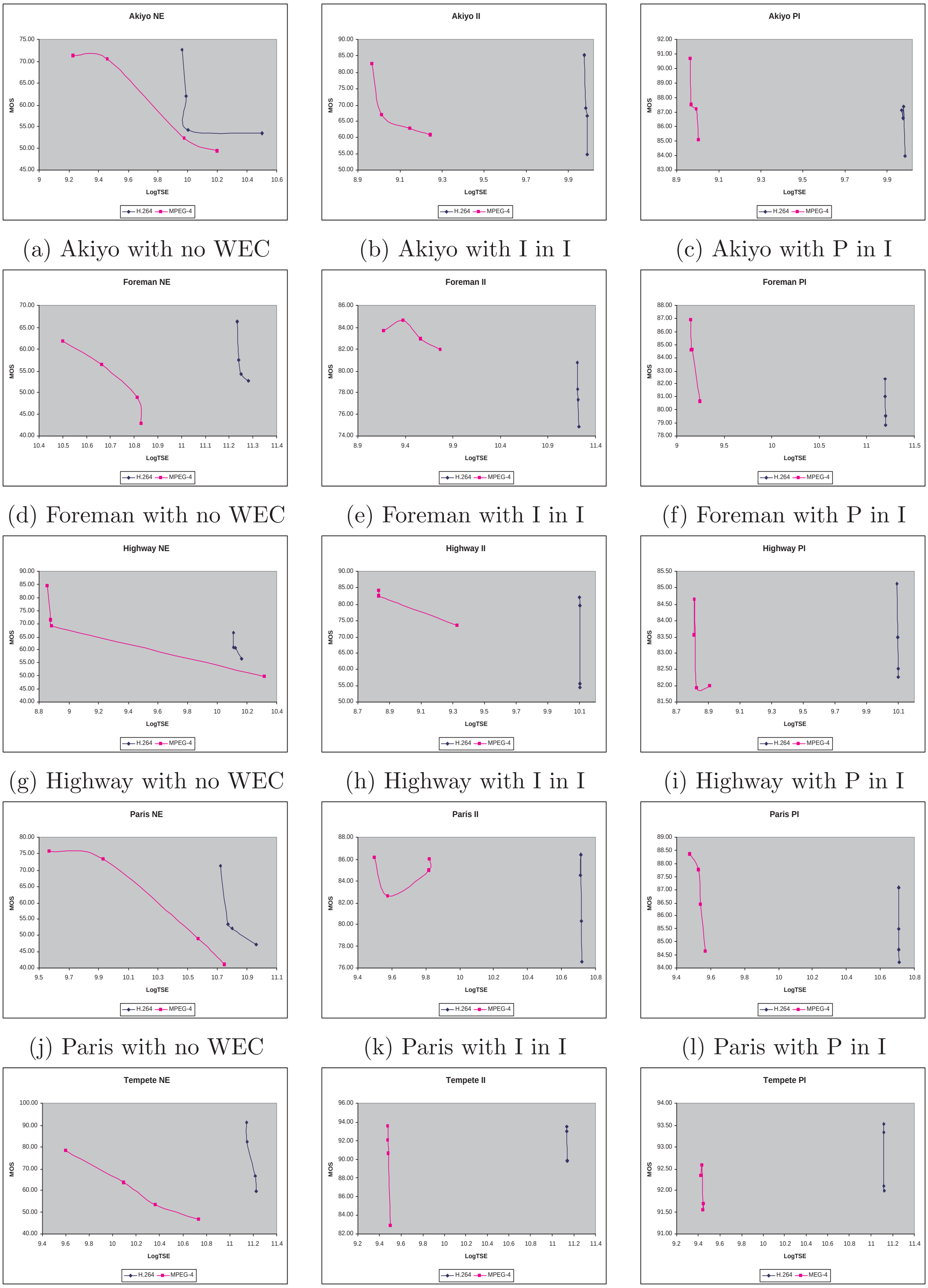
(d) Foreman with no WEC (e) Foreman with I in I (f) Foreman with P in I

(g) Highway with no WEC (h) Highway with I in I (i) Highway with P in I

(j) Paris with no WEC (k) Paris with I in I (l) Paris with P in I

(m) Tempete with no WEC (n) Tempete with I in I (o) Tempete with P in I

Figure 6.9: The plots for the WEC algorithms for each video. The MOS values for the two codecs H.264 and MPEG-4, can be compared using these plots.



(a) Akiyo with no WEC

(b) Akiyo with I in I

(c) Akiyo with P in I

(d) Foreman with no WEC

(e) Foreman with I in I

(f) Foreman with P in I

(g) Highway with no WEC

(h) Highway with I in I

(i) Highway with P in I

(j) Paris with no WEC

(k) Paris with I in I

(l) Paris with P in I

(m) Tempete with no WEC

(n) Tempete with I in I

(o) Tempete with P in I

Figure 6.10: The plots for the WEC algorithms show the video physical errors differences. MPEG-4 can be noted to have lower error for similar MOS values.

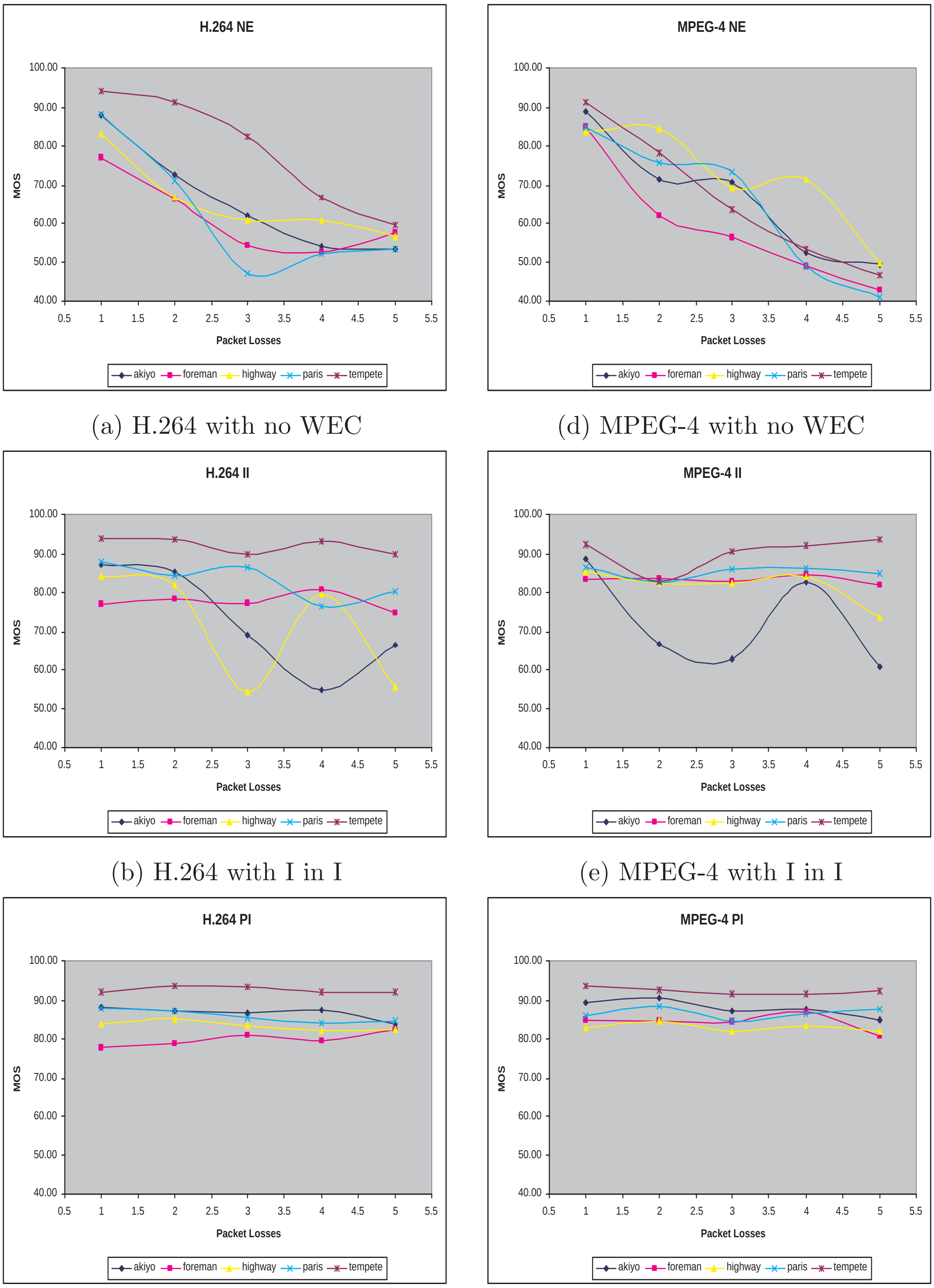
we consider the actual error in the processed videos. For a closer view, let us look at the plots in Fig. 6.10 where the MOS values are plotted against the log TSE values. The figure is similar to Fig. 6.9, where the first column represents no WEC, and the center and last column representing the I-frame and P-frame WEC algorithm implementations respectively.

From all the plots of Fig. 6.10, we see that MPEG-4 gives the resulting video that has a much lesser physical error (log TSE) than H.264. This is expected as H.264 is built to withstand heavy compression losses and give good compression performance while it is susceptible to transmission losses and breaks down when heavy transmission losses occur. MPEG-4, on the other hand, is built for scalability and is known for its transmission loss control (see Section 4.6). However, it is interesting to observe that in almost all the cases of no WEC (except in case of *Highway*), H.264 encoded video obtained similar MOS values to those of MPEG-4. We can conclude from this observation that H.264 gives a perceptually enhanced performance. However, this conclusion does not hold when WEC is applied. Nevertheless, MPEG-4 produces lower error in terms of log TSE than H.264 even in the case of both WEC algorithm implementations. Except for *Foreman* video, the MOS values for the 2 WEC algorithms are similar, but in case *Foreman*, MPEG-4 exhibits higher perceptual quality values.

6.2.4 Content Variation

Even though there is a substantial variation due to the content of the video with regard to the motion and its frequency information, due the constraints of space, time, and conformance of ITU-T standards, we did not experiment with the video content in greater detail. However, we did vary the source video in the limited freedom we had, and the five videos that we chose had different video content. Fig. 6.11 shows the plots of MOS values obtained for different videos in separate WEC algorithm implementations. The first row, (a) and (d) show the MOS values of different videos with varying packet loss percentages both for H.264 and MPEG-4 respectively in case of no WEC algorithm implementation. The second and third rows, (b) and (e), (c) and (f), show the same for I-frame and P-frame WEC algorithm implementations respectively.

We observe that the *Tempete* video obtained the highest MOS values for the cases of



(a) H.264 with no WEC

(d) MPEG-4 with no WEC

(b) H.264 with I in I

(e) MPEG-4 with I in I

(c) H.264 with P in I

(f) MPEG-4 with P in I

Figure 6.11: The plots for each WEC algorithm implementation with each codec. Each plot compares the MOS values over different video contents over the source videos.

H.264 and MPEG-4 (except when no WEC algorithm was used in case of MPEG-4). This may be due to the content of the video and slow zooming camera motion, which helped achieve better compression performance for both the codecs. In case of other videos, we do not notice any one video to exhibit a different trend.

All the videos follow similar trends of obtaining decreasing MOS scores for increasing packet losses for both codecs when no WEC algorithm is implemented. However, this reduction in MOS scores decreases and the curves tend to “flatten out” as the I-frame and P-frame WEC algorithms are subsequently implemented. We also observe some erratic behavior of *Akiyo* and *Highway* videos for H.264 and *Akiyo* video for MPEG-4 when the I-frame WEC algorithm is implemented. This may be because of the way packet losses affected these videos and the chrominance fluctuations obtained due to I-frame WEC algorithm implementation. As explained in Chapter 3, when the packet loss affected areas in a video are concealed, a localized scaling operation is performed to match the luminance and the chrominance of the concealed part with those of surrounding areas. However, in case of *Akiyo* and *Highway* videos, the content includes wide areas of not only low frequency luminance, but also spikes in their color histograms. This may sometimes lead to a color mis-match in temporal domain right at the transition to every I-frame (where WEC is implemented) thereby creating an effect similar to the temporal flicker. This however, is eliminated by having a reference to the subsequent P-frame, which temporally spreads the localized scaling WEC operation. The performance enhancement due to P-frame WEC algorithm implementation over the I-frame WEC algorithm can be observed by comparing the plots in Fig. 6.11(b) and (e) with (c) and (f) respectively.

6.3 Summary

A psychophysical experiment was performed to evaluate the performance of the proposed WEC algorithms for different packet loss percentages and video contents and to verify their codec independency. This chapter explained the aims and the goals of the experiment, along with the it's design structure, the source videos used, the generation of the sequences, and the testing methodology. The results of the experiment were categorically analyzed both with respect to varying packet losses and the amount of error introduced in them due to these transmission losses.

The conclusions drawn from the results of the experiment are well brought out by the data analysis as the following: The performance of the implemented WEC algorithms was observed to be $(no\_WEC) < (I\_WEC) < (P\_WEC)$. This phenomenon was true irrespective of the codec used, the source video, and percentage of packet loss. It was also observed that even though the I-frame WEC algorithm implementation reduced the amount of transmission defects significantly, it received lesser perceptual quality scores (MOS values) when compared to the P-frame WEC algorithm implementation. Based on the data analysis, we conclude that P-frame WEC algorithm is successful in giving a relatively constant perceptual quality output. The curves in Fig. 6.6 lead us to conclude that the P-frame WEC algorithm maximizes the MOS values by minimizing the error in the error concealed sequences. The dual-stimuli display helped us in observing that the MOS values for a given sequence depend on the sequence that is played beside it.

The constant perceptual quality operation of the P-frame WEC algorithm could be verified by the performance enhancements it achieves for increasing packet loss percentages, i.e., the higher the packet loss percentage, the better the WEC algorithm performs. Although the performance of the algorithms was weakly dependent on the content of the video and the codec used for compression, we believe that the effects of compression intervened with the MOS values obtained. Some of the differences in the trends could be explained by a combination of the content of the video and its compression (dependent on the codec). However, the variation of the WEC performance with the variation in amount of compression require further experimentation and is an open research area.

Chapter 7

Conclusions

In this chapter, we discuss the important conclusions of the dissertation work and provide its principal contributions. The set of WEC algorithms that were proposed, along with the video implementations, the theoretical framework, and the psychophysical analysis, could be used in a variety of applications including and not limited to image and video transmission and storage, information compression, coding and retrieval, and in low bit rate, SDTV, and HDTV applications. Some of the possible future directions in the areas of storage, SDTV/HDTV applications, and scalable coding have also been provided towards the end of this chapter.

7.1 Principal Contributions

The work developed a set of a novel error concealment techniques based on watermarking. The generated watermark is a low resolution version of the frame itself and helps recover the information losses that occurred during the transmission of the video. The binary watermark could be a halftoned version or an encoded bit stream of the low resolution version of the frame. The implementation in case of color planes and varied channels losses show the enhanced performance of the developed algorithms. We also developed a low bit error rate informed watermarking scheme that embeds a copy of the watermark detector inside the encoder. The scale factors as per the coefficients were increased based on minimizing the detector's BER performance as well maximizing the embedded frame quality.

Furthermore, we proposed different spatial video implementations, two of the key techniques being embedding the I-frame reference in itself and embedding the subsequent P-frame in the current I-frame. The idea of the latter technique is spawned from the concepts of constant subjective quality preferences. The work also proposed the combined spatio-temporal watermark embedding technique based on 3D-DCT. A gray level reference of the watermark is embedded in the volume cuboid element and it was shown that higher levels of error concealment could be achieved using this approach.

We then looked at an information theoretic approach to the watermark-based error concealment algorithms. The approach initially required an analysis of the performance of the watermark detector and calculation of the probability of error. Based on the overall estimate of the distortion due to this probability of error, the R-D performance of the WEC algorithms has been analyzed and compared to substantial packet loss scenarios.

The subjective quality increment due to WEC over conventional low bit rate codecs such as MPEG-4 and H.264 was evaluated by conducting a psychophysical experiment. The experiments also verifies the codec independent operation of WEC along with evaluating the variation in quality of the compressed videos due to varying channel loss rates. The two spatial implementations of WEC discussed in chapter 4 along with the baseline error concealment in the codecs are used in the experiment for comparative quality assessment.

7.2 Future Directions

The algorithms developed in this dissertation could be used in various applications. However, certain modifications would be required to the proposed WEC algorithms based on the application that they would serve. Some of these feasible application-based modifications, along with new additions and possible improvement driven avenues are explored herein.

7.2.1 MCTF and Scalable Coding Extensions

One plausible extension to the proposed WEC algorithms is towards scalable coding where the time differences of consecutive frames are obtained. The temporal domain difference essentially gives the motion difference between the two frames. A DCT transform between the two frames would evaluate the energy in the motion. Based on the energy distribution of the motion, we can then decide the low motion areas and the high motion areas by designing and implementing a low pass filter and a high pass filter over this motion estimated time difference [120]. This process is also referred to as the motion compensated time filtering (MCTF) and typically is implemented in scalable coding and compression [103].

For the application of error concealment, the low pass filtered output of the MCTF could be considered ideal. This is based on the conclusions that are drawn from previous subjective testing that the humans tend to observe the motion areas more critically than the low motion areas. Since the low pass filtered output of the MCTF is the areas with little or no motion, we can choose to embed the watermark in these coefficients. The embedded reference would be then added to the high pass filtered output (the high motion areas) and the DCT coefficient image is brought back to the time domain. The watermark embedded original could be obtained by adding the time difference frame to the intra-encoded frame. The watermark detection and the error concealment operations are similar to the proposed WEC algorithms in Chapter 3.

7.2.2 Tackling Frame Loss Scenarios

Another feasible future direction would be to design a WEC algorithm that is equipped with tools for handling heavy losses such as frame losses. One way of achieving this would be to embed a copy of the reference, either P-frame reference or an I-frame reference, in multiple host images. For example, if the reference for a third subsequent frame is embedded both in the current frame and the sixth subsequent frame (both forward and backward reference embedding), the chance of recovering the lost frame increases. Another way of tackling heavy frame losses would be to use higher order temporal interpolations and extrapolations similar to view morphing. These techniques, though are computationally intensive, result in a substantially higher and smoother quality video at the end user.

7.2.3 Storage Applications

The WEC algorithms proposed here, with minor modifications, could be extended for usage in applications where the compressed stored video encounters packet losses, that are similar to video transmission scenarios. Typical video storage processes involve compression and/or conversion of the video into a lower resolution (for example, from HDTV to SDTV). Therefore, apart from the memory allocation and retrieval packet losses, the recovered video also experiences compression and conversion losses in combination with the packet losses. This scenario is identical to the case of transmission losses in case of compressed video. Therefore, WEC algorithms that are proposed here could be applied to storage loss scenarios with little or no modification. However, the storage error concealment performance and the WEC parameters' variation with the loss variables still needs to be researched.

7.2.4 Perceptual Evaluation of LBR Transmission Losses

An important segment of this work is the subjective quality evaluation of videos that are affected with transmission losses. Although there are works in literature that aim at subjectively evaluating the compression defects (either their annoyance, detection thresholds, or the quality) and taking steps to design perceptual quality metrics [94], [101], [114], [119], not much has been done with regard to subjectively evaluating the transmission loss defects. With the experiment described in Chapter 6 as the basis, we can set up a series of experiments to evaluate and assess the effects of transmission packet losses on low bit rate (LBR) encoded videos. The eventual goal of this evaluation would be to develop a perceptual quality metric that assesses the video based on the annoyance created by the transmission defects, both with and without the application of error concealment algorithms.

Appendix A

Display Characterization

A.1 Display Gamma

We measured the display gamma of 17" Dell E172FP active matrix TFT LCD flat panel display using two different instruments: (1) Minolta LS110 luminance meter and (2) Photo Research PR650 spectroradiometer and Ocean Optics USB2000 spectroradiometer. Detailed description about the measurement methodology and the curve fitting procedure are given here.

In the first measurement, the display system under test was the combination of the graphic card and the LCD display, as shown in Fig. A.1. The software program used in this measurement varied the 8-bit RGB nominal intensity values to control the graphic card and kept the default color lookup table settings F(R), F(G) and F(B) from the manufacturer. In addition, the LCD display color mode was set to 24-bit colors. RGB nominal values were varied to produce pure white, pure red, pure green and pure blue patches for measurement in luma, red, green and blue channels, respectively. The display luminance (cd/m^2) was then measured using the Minolta LS110 luminance meter. The luminance meter head was positioned perpendicular to the display surface. After measurement, the measured data from each channel was fitted using the gamma function:

$$Y = b + m \cdot x^{\gamma}, \tag{A.1}$$

where b is the offset, m is the multiplicative gain and $i\gamma$ is the gamma constant. The fit was calculated using the MATLAB functions `nlinfit` and `nlpredci` with 95% confidence interval. Figure A.2 shows the measurement data and fitted curves. Table A.1 shows the fitted parameters.

The display system under test followed the same scheme as Fig. A.1 in the second measurement. However, the software program and the graphic card used in this measurement were different from those in the previous measurement. The software program

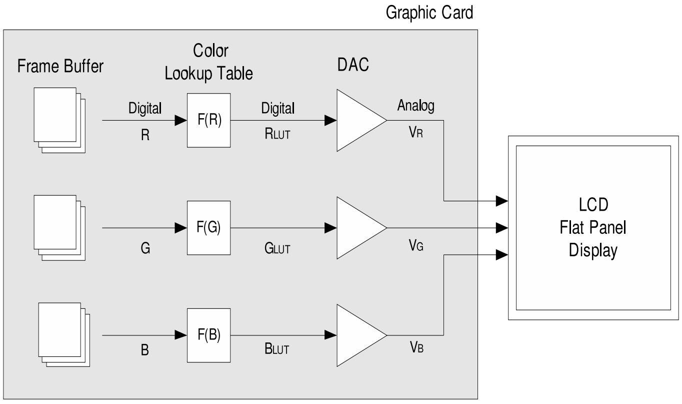


Figure A.1: Display system of graphic card and LCD flat panel display. In general, software programs can control the graphic card by setting RGB nominal intensity values and color lookup tables F(R), F(G) and F(B). The digital-to-analog converter (DAC) translates the digital signals to analog voltages. This figure is adapted from [121] and modified.

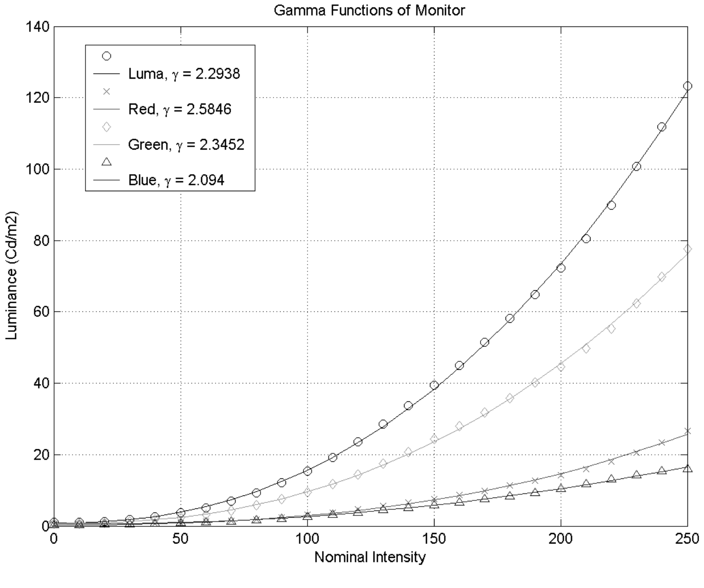


Figure A.2: Display gamma measurement using Minolta LS110 luminance meter. Discrete points represent measured luminance values. Fitted curves are shown as well. All nominal intensities were varied from 0 to 250 with step size of 10.

Table A.1: Fitted parameters in the measurement using Minolta LS110 luminance meter.

	Luma	Red	Green	Blue
b	0.82646	0.57678	0.74788	0.25734
m	3.8282e-4	1.5939e-5	1.7956e-4	1.5531e-4
	2.2938	2.5846	2.3452	2.094

used was the Psychophysics Toolbox developed by D. Brainard et al [122]. This Matlab toolbox has the capability to transparently control the graphic card DAC input signals R$_{LUT}$, G$_{LUT}$, B$_{LUT}$. In addition, the graphic cards DAC has a 10-bit resolution. For this reason, the maximum nominal intensity value was set to 1023 in this measurement. 30 different nominal intensity values in the range between 34 and 1023 were used.

The measurement was conducted using two different models of spectroradiometers: Photo Research PR650 and Ocean Optics USB2000. The spectral radiance distribution, γ, at a certain nominal intensity was taken by averaging the measured data from these two spectroradiometers. At each nominal intensity level, the spectral radiance distribution was measured from 380nm to 780nm at 4nm step. After measurement, XYZ tristimulus values were calculated by evaluating the following integrals:

$$X = K_m \qquad Y = K_m \qquad Z = K_m \tag{A.2}$$

where $K_m = 683(cd\cdot sr/W)$ is the maximum photopic luminous efficacy constant and are CIE1931 color matching functions. It should be noted that when calculating tristimulus values Eq. (A.2) should be replaced by Eq. (A.3) because it is not possible to get continuous spectral radiance distribution γ.

$$X = K_m \qquad Y = K_m \qquad Z = K_m. \tag{A.3}$$

MATLAB functions `nlinfit` and `nlpredci` with 95% confidence interval were used for the curve fitting procedure. The Y tristimulus values were plotted versus nominal intensities in Fig. A.3. It can be seen from the figure that the highest three nominal intensities in each channel are excluded from curve fitting procedure because they exhibit saturation phenomena. Table A.2 shows the fitted parameters.

Comparison between fitted parameters in Tables A.1 and A.2 shows that b and m parameters in Table A.1 are all higher than those in Table A.2. One of the reasons is that the ambient luminance was not measured and subtracted from the measured luminance data in the measurement using Minolta LS110. This mistake was not discovered until the measurement using Photo Research PR650 and Ocean Optics USB2000. Fortunately, γ parameters in Tables A.1 and A.2 are still fairly close. Therefore, this mistake does not

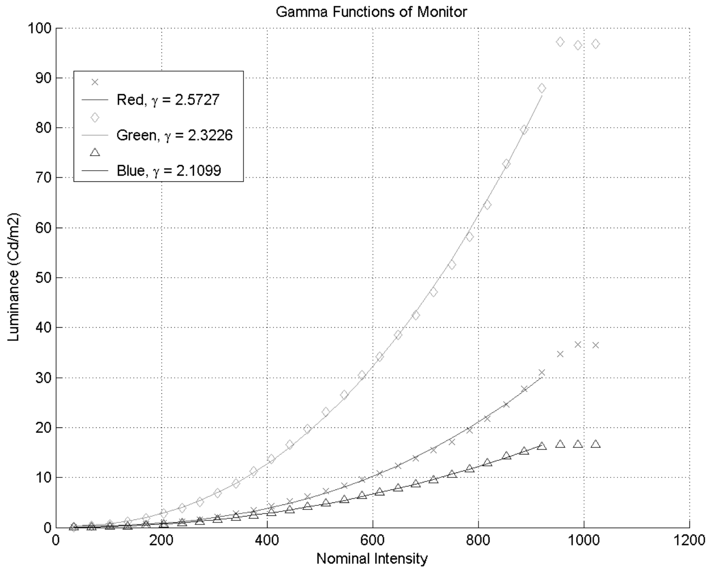


Figure A.3: Display γ measurement using the average from Photo Research PR650 spectroradiometer and Ocean Optics USB2000 spectroradiometer. Discrete points represent measured luminance values. The curve fitting was calculated in the nominal intensity range between 34 and 911. The highest three nominal intensities (955,989,1023) exhibit saturation phenomena and were excluded from curve fitting.

Table A.2: Fitted parameters in the measurement using Photo Research PR650 spectroradiometer and Ocean Optics USB2000 spectroradiometer.

	Red	Green	Blue
b	0.29706	0.28397	0
m	7.0553e-7	1.123e-5	9.1495e-6
	2.5727	2.3226	2.1099

significantly influence the calculation of γ parameters.

It is often the case that for quality monitors the fitted gamma functions in red, green and blue channels all share the same γ parameter. Consequently, it is common to normalize maximum luminance value in each channel and fit a single γ function. However, if a single γ function is used for fitting, the fitted γ function in either measurement is not close to the normalized data in any channel and gives significant errors in predicting the display luminance. For this reason, we decided to use different gamma functions in different channels. One implication of having different gamma functions is that as the nominal intensity increases, the color gamut in the chromaticity coordinate will vary. This variation will be shown in the next section.

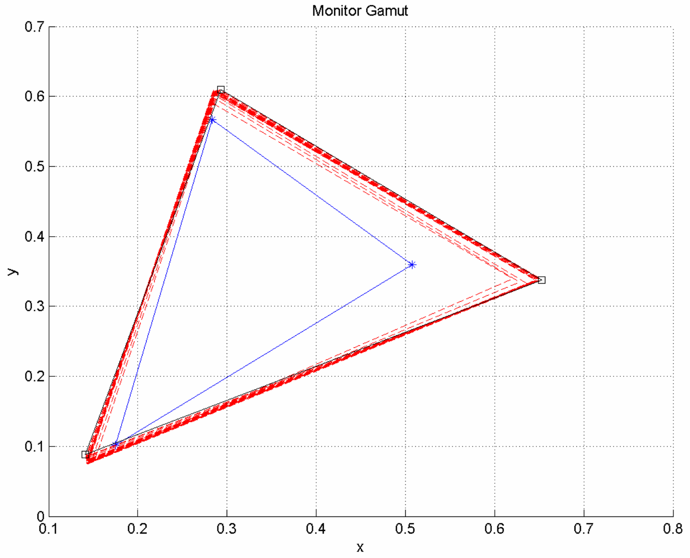


Figure A.4: Display gamuts in xy chromacity coordinate. The innermost blue triangle with asterisk (*) end points represents the display gamut at nominal intensity 34. The black triangle with square (□) end points represents the display gamut at nominal intensity 921. The display gamuts at nominal intensities between 34 and 921 are shown in red dashed line triangles.

A.2 Display Gamut

In order to check the consistency of display gamut as the nominal intensity varies, the xyz chromacity coordinate values were calculated from XYZ tristimulus values in Section A.1 by using the following equations:

$$\begin{aligned} x &= \frac{X}{X+Y+Z} \\ y &= \frac{Y}{X+Y+Z} \\ z &= \frac{Z}{X+Y+Z}. \end{aligned} \tag{A.4}$$

All display gamuts measured in the nominal intensities between 34 and 921 are drawn in the xy chromacity coordinate and shown in Fig. A.4. From the figure, we can observe the variation in display gamut with the variation in the nominal intensity. In particular, the display gamut at nominal intensity 34 deviates maximum from display gamuts at other nominal intensities. We believe that this variation is due to different γ parameters in red, green, and blue channels of the measured LCD display.

Appendix B

Video Quality Experiment Instructions

Before the subject arrives:

1. Log in to the server.

2. Make sure that NumLock is activated.

3. Double-click on the Video Quality Experiment icon.

4. Click on start.

After the subject arrives:

Sit the subject in the chair, centered in front of the LCD monitor. The subject should be adjusted backward or forward to get a distance of 80cm from the screen of the LCD monitor. The most comfortable position for the subject tends to be leaning forward slightly with forearms or elbows on the table.

Read the following instruction to the subject:

1. "This study is concerned with evaluating (rating) the visual quality of videos and their effect on human viewers. We are not concerned with the content of the videos. We are interested in whether or not you see any differences in visual quality in the videos that we will show, which one you feel has a higher quality, and how you evaluate the quality of the videos.

2. "First, a set of reference video sequences will be shown to you. These clips will be of good visual quality. These are shown to set the scale for judging video quality. The best video in this set will be assigned a quality value of 100. Nevertheless, there may be some impairments and defects in these sample clips as well. Once the reference clips are shown to make you aware of the measure of visual quality,

a set of sample clips are shown to give a feel of the different kinds of errors and their ranges, followed by multiple trials showing different video clips. You have to observe and rate their quality based on the reference clips as reference.

3. "Prior to each trial, you will look at the center of the screen of the video monitor. You may move your eyes during the presentation of the clip. You will be presented with two video images on each trial, one on the LEFT and the other on the RIGHT. The video images shown on both sides are the same. Each trial will last 10 seconds. The distance from the monitor to your eyes is important during the presentation. Try not to lean forward or backward. The indicator to your left shows the distance at which your eyes should be.

4. "Boxed spaces will be available on the LCD monitor at the bottom of the LEFT and the RIGHT videos to input your rating for each video in every trial. Do not spend a lot of time thinking about your responses. We want to know your initial impressions.

5. "You have to indicate the quality of both LEFT and RIGHT images with reference to the quality of the best reference clip which has a quality of 100. Indicate your evaluation of both the LEFT and RIGHT clip qualities by entering values in proportion to the quality in the corresponding spaces given on the response screen. For example, if you found the quality of the left clip to be twice as good as the reference clip, then enter 200 in the space marked LEFT, and if you found the quality of the right clip to be half as good as the reference clip, then enter 50 in the space marked RIGHT. If you find the videos to be exactly as good as the reference video, enter the value of 100.

6. "You will be indirectly indicating whether you noticed any difference in the quality of the LEFT and the RIGHT video clips in the region of the image at any time during the clip by either assigning the same rating or a different rating to the LEFT and the RIGHT videos.

7. "If you assign a different value to each of the LEFT and the RIGHT video clips, you are instructed to indicate a higher value to the video which you find to be of higher quality, and a lower value to the other video. For example, if you find the LEFT video clip to have a higher quality than the RIGHT video clip, then you should assign a higher value to the LEFT video clip than the value you assign to the RIGHT video clip.

8. “Once you indicate your quality values, you have completed the evaluation for that trial. You will then be shown another pair of LEFT and RIGHT video clips after which you will respond to the same questions. The computer will start showing the next video clip as soon as you click on the <NEXT> button. Please let the experimenter know if you have any questions with regard to the procedure.”

9. [Show the original clips by clicking on the <REFERENCES> button. Watch five video clips by clicking on the <NEXT> button after each video clip. The program allows the subject to go back and forth by clicking on the <BACK> and the <NEXT> buttons, respectively. At the end of the fifth video, a <SAMPLES> button will appear where the <NEXT> button was previously located. Do not hit on the <SAMPLES> button yet. Let the subjects look at the reference videos and get a feel of what the quality value of 100 looks like.]

10. [Click on the <NEXT> button to proceed with the rest of the reference clips.]

11. “Do you now have a good idea of what the reference rating of 100 looks like? Do you have any questions?

12. “You will next see a set of five sample video clips. These clips are presented to give you a reference to set the scale for rating the video clips. The defects and impairments that you see in these clips extend from very light to heavy. Therefore, some of the videos are of very good quality and some others are of very poor quality. Nonetheless, you might find a better quality video clip or a worse quality video clip in the main experiment.”

13. [Show the sample clips by clicking on the <SAMPLES> button. Watch the five sample video clips by clicking on the <NEXT> button after each clip. The program allows the subject to go back and forth by clicking on the <BACK> and the <NEXT> buttons, respectively. At the end of the fifth video, a <PRACTICE> button will appear where the <NEXT> button was previously located. Do not hit on the <PRACTICE> button yet. Let the subjects look at the sample videos and set a scale for rating the video quality.]

14. “Before we start the experiment, you will have five practice trials to be sure that you understand the task. Once you finish assigning your responses to each trial, you can go to the next trial by clicking on the <NEXT> button. You will respond in these trials just like you will in the main experiment. You can choose to go the main experiment by clicking on the <MAIN> button at time during the practice session. Remember that you have to indicate your responses to both the videos

based on your preference of the LEFT and the RIGHT videos and the reference of 100. Remember to press <NEXT> after entering the quality values. We will not use the data from the practice trials, so do not be concerned if you make a mistake here.

15. [Start the practice trials by hitting the <PRACTICE> button.]

16. "Do you have any questions?

17. "You can take a break at any time by entering your answers for the most recent video, but waiting to hit <NEXT> until you are ready to go on. You should stop if you are confused about what to do, if you realize you have entered data incorrectly, or if you need a break. You cannot stop the video from playing or go back and fix the data from a previous clip after you hit <NEXT>. If, by accident you enter some wrong responses, watch the next video and then tell the experimenter what your responses should have been. We will go back and fix these later.

18. "There are 150 clips in the experiment and it takes approximately 45 minutes to complete, if you do not take any breaks.

19. "Do you have any questions?

20. "At the end of the experiment the experimenter will ask a few questions. Start the experiment by hitting the <MAIN> button. Then enter your first and last names and hit <NEXT>. Finally, when you are ready to start the experiment, hit <START> button.

21. [Start the experiment.]

22. [At the end of the experiment, ask the following questions and write down the answers:]

    - "How would you describe the differences in quality in the video clips that you saw?
    - "Did these clips differ in more than one feature? If so, what features varied?
    - "In your judgement, what made some of the videos better than others?"

Bibliography

[1] Y. Wang and Q.-F. Zhu, “Error control and concealment for video communications: a review,” *Proc. IEEE, Special Issue on Multimedia Signal Processing*, vol. 86, no. 5, May 1998, pp. 974-997.

[2] K. Stuhlmuller, N. Farber, M. Link, and B. Girod “Analysis of video transmission over lossy channels,” *Journal on Selected Areas in Communications*, vol. 18, no. 6, June 2000, pp. 1012-1032.

[3] D. Wu, Y.T. Hou, and Y.-Q. Zhang, “Transporting real-time video over the internet: Challenges and approaches,” *Proc. IEEE*, vol. 88, no. 12, December 2000, pp. 1855-1875.

[4] R. Schafer and T. Sikora, “Digital video standards and their role in video communications,” *Proc. IEEE*, vol. 83, no. 6, June 1995, pp. 907-924.

[5] T. Ebrahimi, E. Reusens, and W. Li, “New trends in very low bitrate coding,” *Proc. IEEE*, vol. 83, no. 6, June 1995, pp. 877-891.

[6] G. Konklin, “Video coding for streaming media delivery on the internet,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 3, March 2001, pp. 269-281.

[7] A. Helal, L. Sampath, K. Birkett, and J. Hammer, “Adaptive delivery of video data over wireless and mobile environments,” *Intl. Journal of Wireless Communications and Mobile Computing*, vol. 3, March 2003, pp. 23-36.

[8] J. Cabrera, A. Ortega, and J.I. Ronda, “Stochastic rate-control of video coders for wireless channels,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 12, no. 6, June 2002, pp. 496-510.

[9] C.Y. Hsu, A. Ortega, and M. Khansari, “Rate control for robust video transmission over burst-error wireless channels,” *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 5, May 1999, pp. 1-18.

[10] K. Fukuda, N. Wakamiya, M. Murata, and H. Miyahara, “QoS mapping between user’s preference and bandwidth control for video transport,” *Proc. Intl. Workshop on Quality of Service*, New York, USA, May 1997, pp. 291-302.

[11] D. Wu and R. Negi, “Effective capacity: A wireless link model for support of quality of service,” *IEEE Trans. on Wireless Communications*, vol. 2, September 2002, pp. 630-643.

[12] J. Ridge, F.W. Ware, and J.D. Gibson, “Permuted smoothed descriptions and refinement coding for images,” *IEEE Journal on Selected Areas of Communication*, vol. 18, no. 6, June 2000, pp. 915-926.

[13] Y. Wang, S. Wenger, J. Wen, and A.K. Katsaggelos, “Error resilient video coding techniques,” *IEEE Signal Processing Magazine*, vol. 17, no. 4, July 2000, pp. 61-82.

[14] R. Singh, A. Ortega, L. Perret, and W. Jiang, “Comparison of multiple description coding and layered coding based on network simulations,” *Proc. SPIE, Image and Video Communications and Processing Conf.*, San Jose, CA, USA, January 2000, pp. 929-939.

[15] J. Kim, R. Mersereau, and Y. Altunbasak, “Error-resilient image and video transmission over the internet using unequal error protection,” *IEEE Trans. on Image Processing*, vol. 12, no. 2, February 2003, pp. 121-131.

[16] A.A. Alatan, M. Zhao, and A.N. Akansu, “Unequal error protection of SPIHT encoded image bit streams,” *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, June 2000, pp. 814-818.

[17] M.C. Hong, L. Kondi, H. Scwab, and A.K. Katsaggelos, “Video error concealment techniques,” *Signal Processing: Image Communications, Special Issue on Error Resilient Video*, vol. 14, no. 6-8, May 1999, pp. 437-492.

[18] H. Wang and A. Ortega, “Robust video communication by combining scalability and multiple description coding techniques,” *Proc. SPIE, Image and Video Communications and Processing Conf.*, San Jose, CA, USA, January 2003, pp.111-124.

[19] D.S. Turaga, and T. Chen, “Model-based error concealment for wireless video,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 12, no. 6, June 2002, pp. 483-495.

[20] C.B. Adsumilli and Y.H. Hu, “Adaptive wireless video communications: Challenges and approaches,” *Proc. Intl. Packet Video Workshop*, Pittsburg, PA, USA, April 2002.

[21] A.K. Katsaggelos, F. Ishtiaq, L. P. Kondi, M.-C. Hong, M.Banham, and J. Brailean, “Error resilience and concealment in video coding,” *Proc. European Signal Processing Conference (EUSIPCO-98)*, Greece, September 1998, pp. 221-228.

[22] Y. Wang, J. Osterman, and Y.-Q. Zhang, *Video Processing and Communications*, Printice Hall, New Jersey, USA, 2002.

[23] S.B. Wicker, *Error Control Systems of Digital Communication and Storage*, Printice Hall, New Jersey, USA, 1995.

[24] M. Buckley, M. Ramos, S. Hemami, and S. Wicker, "Perceptually-based robust image transmission over wireless channels," *Proc. IEEE Intl. Conf. on Image Processing*, vol. 2, Vancouver, Canada, September 2000, pp. 128-131.

[25] C.B. Adsumilli and Y.H. Hu, "A dynamically adaptive constrained unequal error protection scheme for video transmission over wireless channels," *Proc. Multimedia Signal Processing Workshop*, St. Thomas, Virgin Islands, USA, December 2002, pp. 41-44.

[26] B. Girod and N. Farber, "Feedback-based error control for mobile video transmission," *Proc. IEEE*, vol. 87, no. 10, October 1999, pp. 1707-1723.

[27] M. Barni and F. Bartolini, *Watermarking Systems Engineering: Enablong Digital Assets Security and Other Applications*, Signal Processing and Communications Series, Marcel Decker Inc., New York, 2004.

[28] X. Chen, *Transporting Compressed Digital Video*, Kluwer Academic Publishers, 2002.

[29] S. Lin and D.J. Costello, *Error Control Coding: Fundamentals and Applications*, Prentice-Hall Inc., New Jersey, 1983.

[30] J. Chen, U.-V. Koc, and K.J. Liu, *Design of Digital Video Coding Systems*, Signal Processing and Communications Series, Marcel Decker Inc., New York, 2002.

[31] ITU Recommendation BT.500-8, "Methodology for the subjective assessment of the quality of television pictures," 1998.

[32] L. Hanzo, P. Cherriman, and E.L. Kuan, "Interactive cellular and cordless video telephony: State-of-the-art system design principles and expected performance," *Proc. IEEE*, vol. 88, no. 9, September 2000, pp. 1388-1413.

[33] L. Hanzo, "Bandwidth-efficient wireless multimedia communications," *Proc. IEEE*, vol. 86, no. 7, July 1998, pp. 1342-1382.

[34] L. Hanzo, C.H.Wong, and P.Cherriman, "Channel-adaptive wideband wireless video telephony," *IEEE Signal Processing Magazine*, vol. 17, no. 4, July 2000, pp. 10-30.

[35] K.N. Ngan, C.W. Yap, and K.T. Tan, *Video Coding for Wireless Communication Systems*, Signal Processing and Communication Series, Marcel Dekker Inc., New York, 2001.

[36] G. Cheung and A. Zakhor, "Joint source/channel coding of scalable video over noisy channels," *Project Report*, University of California at Berkeley, 2000.

[37] S. Yang, S. Kittitornkun, Y. H. Hu, T. Q. Nguyen, and D. L. Tull, "Low bit rate video sequence coding artifact removal," *Proc. IEEE Workshop on Multimedia and Signal Processing*, Cannes, France, October 2001, pp. 53-58.

[38] C.B. Adsumilli, C. Vural, and D.L. Tull, "A noise based quantization model for restoring block transform compressed images," *Proc. IASTED Intl. Conf. on Signal and Image Processing*, Honolulu, HA, USA, August 2001, pp. 354-359.

[39] J.D. Villasenor, Y.-Q. Zhang, and J. Wen, "Robust video coding algorithms and systems," *Technical Report*, University of California at Los Angeles, 1998.

[40] J. Konrad, "Visual communications of tomorrow: Natural, efficient and flexible," *IEEE Communications Magazine*, vol. 39, no. 1, January 2001, pp. 126-133.

[41] International Telecommunication Union (ITU)-T, Draft ITU-T Recommendation H.263, *Telecommunication Standardization Sector of ITU*, Source: R.Schaphorst, Rapporteur, July 5, 1995.

[42] P. Cherriman and L. Hanzo, "Robust H.263 video transmission over mobile channels in interference limited environments," *Proc. IEEE Intl. Workshop on Wireless Image/Video Communications*, Loughborough, UK, September 1996, pp. 1-7.

[43] P. Cherriman, E.L. Kuan, and L. Hanzo, "Burst-by-burst adaptive joint detection CDMA/H.263 based video telephony," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 12, no. 5, May 2002, pp. 342-348.

[44] J.Y. Liao and J.D. Villasenor, "Adaptive intra block update for robust transmission of H.263," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 10, no. 1, February 2000, pp. 30-35.

[45] A.M. Tekalp, *Digital Video Processing*, Signal Processing Series, Prentice Hall, New Jersey, USA, 1995.

[46] J. Meierhofer and G. Fankhauser, "Error resilient, tagged video stream coding with wireless data link control, *Technical Report*, Computer Engineering and Networks Lab, Swiss Federal Institute of Technology, 1999.

[47] Q. Zhao, P. Cosman, and L. Milstein, "'Tradeoffs of source coding, channel coding and spreading in frequency selective rayleigh fading channels," *Journal of VLSI Signal Processing*, vol. 30, February 2002, pp. 7-20.

[48] S. Dennett, *The cdma2000 ITU-R RTT candidate submission (0.18)*, chair TR45.5.4, 1998.

[49] M. Grangetto, E. Magli, M. Marzo, and G. Olmo, “Guaranteeing quality of service for image transmission by means of hybrid loss protection,” *Proc. Intl. Conf. on Multimedia and Expo*, vol. 2, Lausanne, Switzerland, August 2002, pp. 469-472.

[50] A.E. Mohr, E.A. Riskin, and R.E. Ladner, “Unequal loss protection: Graceful degradation over packet erasure channels through forward error correction,” *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 7, June 2000, pp. 819-828.

[51] S.L. Regunathan and K. Rose, “Robust video compression for time-varying wireless channels,” *Proc. SPIE, Visual Communications and Image Processing Conf.*, vol. 3653, San Jose, CA, USA, January 1999, pp. 241-248.

[52] M.S. Moore, J.M. Foley, and S.K. Mitra, “A comparison of the detectability and annoyance value of embedded MPEG-2 artifacts of different type, size and duration,” *Proc. SPIE, Human Vision and Electronic Imaging VI*, vol. 3959, San Jose, CA, USA, January 2001, pp. 99-110.

[53] J.G. Kim and M.M. Krunz, “Bandwidth allocation in wireless networks with guaranteed packet loss performance,” *IEEE/ACM Trans. on Networking*, vol. 8, no. 3, June 2000, pp. 337-349.

[54] D. Tse and S. Hanly, “Effective bandwidths in wireless networks with multiuser receivers,” *Proc. IEEE Infocom*, vol. 1, San Francisco, CA, USA, March 1998, pp. 35-42.

[55] Z. Xiong, K. Ramachandran, M.T. Orchard, and Y.-Q. Zhang, “A comparative study of DCT- and Wavelet based image coding,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 9, no. 8, August 1999, pp. 692-695.

[56] D. Qiao and K.G. Shin, “A two-step adaptive error recovery scheme for video transmission over the wireless networks,” *Proc. IEEE Infocom*, vol. 3, Tel Aviv, Israel, March 2000, pp. 1698-1704.

[57] T. Kalkar, “Considerations on watermarking security,” *Proc. IEEE Workshop on Multimedia Signal Processing*, Cannes, France, October 2001, pp. 201-206.

[58] C.B. Adsumilli, M.C.Q. Farias, M. Carli, and S.K. Mitra, “A hybrid constrained unequal error protection and data hiding scheme for packet video transmission,” *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, vol. 5, Hong Kong, April 2003, pp. 680-683.

[59] Y. Liu and Y. Li, “Error concealment for digital images using data hiding,” *Proc. IEEE Digital Signal Processing Workshop*, Hunt, Texas, USA, October 2000.

[60] Y. Shao, L. Zhang, G. Wu, and X. Lin, "Reconstruction of missing blocks in image transmission by using self-embedding," *Proc. Intl. Symposium on Intelligent Multimedia, Video and Speech Processing*, Hong Kong, May 2001, pp. 535-538.

[61] P. Yin, B. Liu, and H.H. Yu, "Error concealment using data hiding," *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, vol. 3, Salt Lake City, Utah, USA, May 2001, pp. 1453-1456.

[62] C.-S. Lu, "Wireless multimedia error resilience via a data hiding technique," *Proc. Intl. Workshop on Multimedia Signal Processing*, St. Thomas, Virgin Islands, USA, December 2002, pp. 316-319.

[63] J. Lee and C.S. Won, "A watermarking sequence using parities of error control coding for image authentification and correction," *IEEE Trans. on Consumer Electronics*, vol. 46, no. 2, May 2000, pp. 313-317.

[64] J. Wang and L. Ji, "A region and data hiding based error concealment scheme for images," *IEEE Trans. on Consumer Electronics*, vol. 47, no. 2, May 2001, pp. 257-262.

[65] F. Bartolini, A. Manetti, A. Piva, and M. Barni, "A data hiding approach for correcting errors in H.263 video transmitted over a noisy channel," *Proc. IEEE Workshop on Multimedia Signal Processing*, Cannes, France, October 2001, pp. 65-70.

[66] K. Munadi, M. Kurosaki, and H. Kiya, "Error concealment using digital watermarking technique for interframe video coding," *Proc. Intl. Technical Conf. on Circuits/Systems, Computers, and Communications*, Phuket, Thailand, July 2002, pp. 599-602.

[67] A. Yilmaz and A.A. Alatan, "Error concealment of video sequences by data hiding," *Proc. Intl. Conf. on Image Processing*, vol. 3, Barcelona, Spain, September 2003, pp. 679-682.

[68] P. Salama, N.B. Shroff, E.J. Coyle, and E.J. Delp, "Error concealment techniques for encoded video streams," *Proc. Intl. Conf. on Image Processing*, vol. 1, Washington, DC, USA, October 1995, pp. 9-12.

[69] J.W. Park and S.U. Lee, "Recovery of corrupted image data based on the NURBS interpolation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 9, no. 10, October 1999, pp. 1003-1008.

[70] H. Sun and W. Kwok, "Concealment of damaged block transform coded images using projection onto convex sets," *IEEE Trans. on Image Processing*, vol. 4, no. 4, April 1995, pp. 470-477.

[71] Y. Wang and Q. Zhu, "Signal loss recovery in DCT-based image and video codecs," *Proc. SPIE, Visual Communications and Image Processing Conf.*, vol. 1605, Boston, MA, USA, November 1991, pp. 667-678.

[72] X. Li and M.T. Orchard, "Novel sequential error-concealment techniques using orientation adaptive interpolation," *IEEE Trans. on Circuits and Systems on Video Technology*, vol. 12, no. 10, October 2002, pp. 857-864.

[73] I. Cox, J. Kilian, F. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. on Image Processing*, vol. 6, no. 12, December 1997, pp. 1673-1687.

[74] M.D. Swanson, M. Kobayashi, and A.H. Tewfik, "Multimedia data-embedding and watermarking technologies," *Proc. IEEE*, vol. 86, no. 6, June 1998, pp. 1064-1087.

[75] J. Buchanan and L. Streit, "Threshold-diffuse hybrid halftoning methods," *Proc. Western Computer Graphics Symposium (SKIGRAPH)*, Whistler, Canada, April 1997, pp. 79-90.

[76] R.W. Floyd and L. Steinberg, "An adpative algorithm for spatial gray-scale," *Proc. Society of Information Display*, vol. 17, no. 2, April 1976, pp. 75-78.

[77] Z. Xiong, M.T. Orchard, and K. Ramachandran, "Inverse halftoning using wavelets," *IEEE Trans. on Image Processing*, vol. 8, no. 10, October 1999, pp. 1479-1483.

[78] M. Mese and P.P. Vaidyanathan, "Look up table (LUT) inverse halftoning," *IEEE Trans. on Image Processing*, vol. 10, no. 10, October 2001, pp. 1566-1578.

[79] M.-Y. Shen and C.-C. Kuo, "A robust nonlinear filtering approach to inverse halftoning," *Journal of Visual Communications and Image Representation*, vol. 12, no.1, March 2001, pp. 84-95.

[80] J.J. Lemmon, "Wireless link statistical bit error model," *Dept. of Commerce (Communication and Information)*, NTIA Report 02-394, 2002.

[81] D.L. Lau and G.R. Arce, *Modern Digital Halftoning*, Marcel Dekker, Oxford, UK, 2001.

[82] H.R. Kang, *Color Technology for Electronic Imaging Device*, SPIE Optical Engineering Press, New York, USA, 1997.

[83] The network simulator (1995), [online], *http://www.isi.edu/nsnam/ns/*, retrieved April 2003.

[84] L.S. Marvel and C.T. Retter, "The use of side information in image steganography," *Proc. Intl. Symposium on Information Theory and Its Applications*, Honolulu, HA, USA, November 2000.

[85] M. Wu, "Multimedia data hiding," *Ph.D. Dessertation*, Princeton University, Princeton, USA, June 2001.

[86] C.B. Adsumilli, M.C. Farias, M. Carli, and S.K. Mitra, "A robust error concealment technique using data hiding for image and video transmission over lossy channels," *IEEE Trans. on Circuits and Systems for Video Technology*, in press.

[87] M.H. Costa, "Writing on dirty paper," *IEEE Trans. on Information Theory*, vol. IT-29, no. 3, March 1983, pp. 439-441.

[88] M.L. Miller, I.J. Cox, and J.A. Bloom, "Informed embedding: Exploiting image and detector information during watermark insertion," *Proc. IEEE Intl. Conf. on Image Processing*, vol. 3, Vancouver, Canada, September 2000, pp. 1-4.

[89] R. Fakeh, A. Ghani, M. Saman, A. Ramli, "Low-bit-rate scalable compression of mobile wireless video," *Proc. IEEE Region 10 Intl. Conf. (TENCON) on Intelligent Systems and Technologies for the New Millennium*, Kaula Lumpur, Malaysia, September 2000, pp. 201-206.

[90] K. Goh, J. Soraghan, T. Durrani, "Multi-resolution based algorithms for low bit-rate image coding," *Proc. Intl. Conf. on Image Processing*, Austin, Texas, USA, November 1994, pp. 285-289.

[91] E.C. Reed and F. Dufaux, "Constrained bit-rate control for very low bit rate streaming video applications," *IEEE Trans. on Circuits and Systems on Video Technology*, vol. 11, no. 7, July 2001, pp. 882-889.

[92] Z. Vranyecz and K. Fazakas, "Very low bit rate video coding methods for multimedia," *citeseer.nj.nec.com/456947.html*

[93] L. Wang, "Rate control for MPEG video coding," *Journal of Signal Processing: Image Communication*, vol. 15, no. 6, March 2000, pp. 493-511.

[94] N. Vasconcelos and F. Dufaux, "Pre and post-filtering for low bit-rate video coding," *Proc. Intl. Conf. on Image Processing*, vol. 1, Santa Barbara, CA, USA, October 1997, pp. 291-294.

[95] F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: A review and a new contribution," *Proc. IEEE*, vol. 83, no. 6, June 1995, pp. 858-876.

[96] A.M. Tekalp, *Digital Video Processing*, Printice Hall, New Jersey, USA, 1995.

[97] ITU-T H.264 and ISO/IEC 14496-10, "H.264 AVC/MPEG-4 Part 10 standard," 2003.

[98] ISO/IEC 14496-2, "MPEG-4 Part 2: Visual advanced simple profile (ASP) coding standard," 2002.

[99] H.264/AVC JM Reference software, [on-line], *http://iphome.hhi.de/suehring/tml/*, retrieved July 2004.

[100] I.E. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia*, Wiley, Stafford, Australia, 2003.

[101] I. Bouazizi, "Estimation of packet loss effects on video quality," *Proc. Intl. Symposium on Communications, Control, and Signal Processing*, Hammamet, Tunisia, March 2004, pp. 91-94.

[102] Y.S. Saw, *Rate-Quality Optimized Video Coding*, Kluwer Academic Publishers, Springer Link, New York, USA, 1999.

[103] M.-T. Sun and A.R. Reibman, *Compressed Video Over Networks*, Signal Processing and Communications Series, Marcel Decker Inc., New York, 2001.

[104] K.A. Peker, A.A. Alatan, and A.N. Akansu, "Low-level motion activity features for semantic characterization of video," *Proc. IEEE Intl. Conf. on Multimedia and Expo*, New York City, NY, USA, August 2000, pp. 801-804.

[105] M. Servais and G. Jager, "Video compression using the three dimensional discrete cosine transform (3D DCT)," *Proc. IEEE Intl. Conf. on Communications and Signal Processing (COMSIG)*, Grahamstown, South Africa, September 1997, pp. 27-32.

[106] Y. Wu, X. Gun, M.S. Kankanhalli, and Z. Huang, "Robust invisible watermarking of volume data using the 3D DCT," *Proc. IEEE Computer Graphics Intl. Conf.*, Hong Kong, July 2001, pp. 359-362.

[107] A. Tefas, G. Louizis, and I. Pitas, "3D image watermarking robust to geometric distortions," *Proc. IEEE Intl. Symposium on Signal Processing and Information Technology*, Cairo, Egypt, December 2001, pp. 229-232.

[108] A.M. Alattar, E.T. Lin, M.U. Celik, "Digital watermarking of low bit rate advanced simple profile MPEG-4 compressed video," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 8, August 2003, pp. 787-800.

[109] J.R. Hernandez, F.P. Gonzalez, and M. Amado, "DCT-domain image watermarking and generalized gaussian models," *Proc. Intl. Workshop on Intelligent Communications and Multimedia Terminals*, Ljubljana, Slovenia, November 1998.

[110] J.R. Hernandez, M. Amado, and F.P. Gonzalez, "DCT-domain watermarking techniques for still images: Detector performance and a new structure," *IEEE Trans. Image Processing*, vol. 9, no. 1, January 2000, pp. 55-68.

[111] L. Hua and J.E. Flower, “A performance analysis of spread-spectrum watermarking based on redundant transforms,” *Proc. IEEE Intl. Conf. Multimedia and Expo*, vol. 2, Lausanne, Switzerland, August 2002, pp. 553-556.

[112] R.C. Reininger and J.D. Gibson, “Distributions of the two-dimensional DCT coefficients for images,” *IEEE Trans. on Communications*, vol. 31, no. 6, June 1983, pp. 835-839.

[113] E.Y. Lum and J.W. Goodman, “A mathematical analysis of the DCT coefficient distributions for images,” *IEEE Trans. on Image Processing*, vol. 9, no. 10, October 2000, pp. 1661-1666.

[114] S. Winkler, “Issues in vision modeling for perceptual video quality assessment,” *Journal of Signal Processing*, vol. 78, no. 2, December 1999, pp. 231-252.

[115] Video Quality Experts Group, “VQEG subjective test plan,” *Techcnical Report*, http://ftp.crc.ca/test/pub/crc/vqeg/, 1999.

[116] A.B. Watson, “Towards a visual quality metric for digital video,” *Proc. European Signal Processing Conf.*, vol. 2, Island of Rhodes, Greece, September 1998.

[117] C.J. Lambrecht and M. Kunt, “Characterization of human visual sensitivity for video imaging applications,” *Journal of Signal Processing*, vol. 67, no. 3, June 1998, pp. 255-269.

[118] J. Lubin, “A human vision system model for objective picture quality measurements,” *Proc. IEE Intl. Broadcasting Conf.*, Amsterdam, The Netherlands, September 1997, pp. 498-503.

[119] V. Algazi, N. Hiwasa, “Perceptual criteria and design alternatives for low bit rate video coding,” *Proc. Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, USA, November 1993, pp. 831-835.

[120] I. Setyawan and R. Lagendijk, “Low bit-rate video watermarking using temporally extended differential energy watermarking (DEW) algorithm,” *Proc. SPIE, Conf. on Security and Watermarking of Multimedia Contents*, San Jose, CA, USA, January 2001, pp. 73-84.

[121] D.H. Brainard, D.G. Pelli, and T. Robson, “Display characterization,” *Encyclopedia of Imaging Science and Technology*, J. Hornak (ed.), Wiley, Stafford, Australia, 2002, pp. 172-188.

[122] [Online], *http://www.psychtoolbox.org/*, retrieved April 2005.