ROBUST VIDEO TRANSMISSION USING DATA HIDING

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OF

THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

AYHAN YILMAZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2003

Approval of the Graduate School of Natural and Applied Sciences

_____

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

_____

Prof. Dr. Mübeccel Demirekler
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____

Assoc. Prof. Dr. A. Aydın Alatan
Supervisor

Examining Committee Members

Prof. Dr. Murat Aşkar _____

Prof. Dr. Mete Severcan _____

Assoc. Prof. Dr. A. Aydın Alatan _____

Assoc. Prof. Dr. Gözde Bozdağı Akar _____

Dr. H. Orkun Zorba _____

ABSTRACT

ROBUST VIDEO TRANSMISSION USING DATA HIDING

Yılmaz, Ayhan

M.Sc., Department of Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Dr. A. Aydın Alatan

September 2003, 69 pages

Video transmission over noisy wireless channels leads to errors on video, which degrades the visual quality notably and makes error concealment an indispensable job. In the literature, there are several error concealment techniques based on estimating the lost parts of the video from the available data. Utilization of data hiding for this problem, which seems to be an alternative of predicting the lost data, provides a reserve information about the video to the receiver while unchanging the transmitted bit-stream syntax; hence, improves the reconstruction video quality without significant extra channel utilization. A complete error resilient video transmission codec is proposed, utilizing imperceptible embedded information for combined detecting, resynchronization and reconstruction of the errors and lost data. The data, which is imperceptibly embedded into the video itself at the encoder, is extracted from the video at the decoder side to be utilized in error concealment. A

spatial domain error recovery technique, which hides edge orientation information of a block, and a resynchronization technique, which embeds bit length of a block into other blocks are combined, as well as some parity information about the hidden data, to conceal channel errors on intra-coded frames of a video sequence. The errors on inter-coded frames are basically recovered by hiding motion vector information along with a checksum into the next frames. The simulation results show that the proposed approach performs superior to conventional approaches for concealing the errors in binary symmetric channels, especially for higher bit rates and error rates.

Keywords: Robust Video Transmission, Data Hiding, Error Concealment, Synchronization, Error Detection, H.263+

ÖZ

BİLGİ SAKLAMA İLE DAYANIKLI VİDEO İLETİMİ

Yılmaz, Ayhan

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez yöneticisi: Doç. Dr. A. Aydın Alatan

Eylül 2003, 69 sayfa

Videonun gürültülü bir kanaldan iletimi, görüntü kalitesini önemli ölçüde düşüren hatalara sebep olmakta ve bu hataların saklanmasını kaçınılmaz kılmaktadır. Literatürde, eldeki verileri kullanarak hatalı kısımların tahminine dayalı bir çok hata düzeltme yöntemi bulunmaktadır. Bu problemin çözümünde, hatalı kısımların öngörüsüne seçenek gibi görünen, bilgi saklama yaklaşımını kullanmak alıcıya görüntü hakkında yedek bilgi sağlarken, iletilecek bitlerin diziminde herhangi bir değişikliğe neden olmamaktadır. Böylece ekstra bir kanal ayarına gerek kalmadan, görüntünün onarılma kalitesi artmaktadır. Önerilen video kodlayıcı-kodçözücü, iletim hatalarının seziminde, tekrar eşzamanlamanın sağlanmasında ve hataların onarılmasında saklı bilgiyi kullanmaktadır. Kodlayıcı tarafında videonun içine görünmez bir şekilde saklanan bilgi, hata düzeltmede kullanılmak üzere kodçözücüde çıkartılır. Çerçeve içi kodlanmış videolardaki kanal hatalarını

düzeltmek için, bloğun ayrıt yön bilgisini saklayan bir konumsal hata onarma yöntemi ile, bloğun bit uzunluğunu saklayan bir eşzamanlama sağlama yöntemi birleştirilmiş ve buna ek olarak, saklanan bilgi ile ilgili bazı eşlik bilgileri de kullanılmıştır. Çerçeve içi kodlanmış videolar ise temel olarak devinim vektörlerinin bazı sağlama bitleriyle birlikte diğer çerçevelere saklanması ile onarılır. İkili bakışımlı kanal hatalarının düzeltilmesi deneylerinin sonuçları, özellikle yüksek bit hızlarında, önerilen yöntemin diğer alışılagelmiş yöntemlerden daha başarılı bir performans sağladığını göstermiştir.

Anahtar Kelimeler: Dayanıklı Video İletimi, Bilgi Saklama, Hata Düzeltme, Eşzamanlama, Hata Algılama, H.263+

To My Mom

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my advisor, Assoc. Prof. Dr. Aydın Alatan for his guidance, encouragement and support in every stage of this research.

I would also like to thank to Prof. Dr. Murat Aşkar for his enlightening discussions and comments at the last phases of the research.

I am also grateful to Serdar İnce, my officemate, for the collaborations in my early research.

This thesis draws a period for my 20-year education in schools. In addition to my teachers, classmates, and roommates over the past two decades, I must mention two persons without whose support I could never accomplish all these. To my high school teachers, Hatice Uça and Muammer Akıncı, I offer sincere thanks for their endless support and unshakable faith in me.

Finally, I would like to express my deep gratitude to all who have encouraged and helped me at the different stages of this work.

And my parents and sisters, I thank them for everything.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURE

LIST OF ABBREVIATIONS

BER	Bit Error Rate
BMA	Boundary Matching Algorithm
BSC	Binary Symmetric Channel
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DWT	Discrete Wavelet Transform
ECC	Error Control Coding
FEC	Forward Error Correction
HVS	Human Visual System
LSB	Least Significant Bit
MB	Macroblock
MFI	Motion Field Interpolation
MSB	Most Significant Bit
MV	Motion Vector
PH	Picture Header
QCIF	Quarter Common Interface Format
QIM	Quantization Index Modulation
RS	Reed-Solomon

CHAPTER 1

INTRODUCTION

Next generation wireless systems promise higher bit rates which can accommodate video transmission to wireless devices. However, wireless transmission is always affected by the environmental (atmospheric or interference of other electronic systems) noise and the transmission of video signals over noisy wireless channels may cause inevitable errors that might severely degrade the visual message. In wireless communication systems, in order to handle such errors, some error concealment techniques have been proposed [1-6]. In three major groups, these techniques try to recover the lost data either by an interaction between the encoder and decoder, as a re-send signal [2,4], or post-processing operations at the decoder to recover lost information [3,5,6], or leaving some extra redundancy at the encoder to minimize the reconstruction error [1].

In the encoder and decoder interactive error concealment techniques, encoder and decoder cooperate, if a backward channel from decoder to encoder is available. Based on the feedback information, source coding parameters, the amount of Forward Error Correction (FEC) bits, and retransmission bandwidth can be changed. However, retransmission leads to decoding delays, which is not desirable in some real-time systems.

Post-processing error concealment techniques use the correlation between the damaged block and its neighboring blocks in the same frame and/or previous frame. These techniques are based on the smooth variation of the intensity values of spatial and temporally adjacent pixels. However, in the regions with sharp edges, a satisfying reconstruction may not be achieved by post-processing operations.

The error concealment techniques in the third group utilize the redundant data on the bit stream, which is added at the encoder side after source coding. Video source can be coded in layers or in multiple descriptions during the source coding and some amount of FEC can be applied. The major drawback of these methods is the increasing transmission overhead.

All these approaches can be merged together by hiding some imperceptible information to be useful during error concealment. During source coding, some information about the video can be embedded into certain parts of the video itself and the decoder can make use of this hidden information in error concealment. In this way, hidden information is not only transmitted through a secret channel from encoder to decoder by “sending back” some lost information, but also alleviates some burden on post-processing.

Hiding some data into a video slightly degrades the visual quality and causes a minor increase in the coding bit rate. On the other hand, the extra hidden information and its small visual loss might be equivalent to decreasing the source bit-rate for obtaining the same visual quality and utilizing error control codes as a result of the bit savings at the encoder.

This radical approach, employing hidden data in video error concealment, is a result of steganography, a new technique for making imperceptible modifications on the media, mostly utilized for copyright protection and other security-based applications [7,8]. It should be emphasized that the hidden information can be transmitted without a significant bit-rate overhead in the bit-stream of the compression standard being used. The standard receivers unaware of such hidden information will be unaffected and decode the bit-stream, successfully (i.e. *backward compatibility* between the bit-streams and conventional decoders).

Without the utilization of data hiding approach, the transmission of equivalent information in a bit-stream requires an extra bit-rate, as well as some modifications in all the (standard compatible) receivers to understand or discard such a message. Obviously, the price, one pays for this additional gain due to data hiding, is an increasing complexity at the receiver to decode the hidden information and a small loss in visual quality due to the embedded signal.

1.1 Problem Definition

All state-of-the-art image and video codecs are block-based and possible bit-errors usually destroy the data only in a single block or even all the blocks in the rest of the row of macroblocks (slice). These block errors decrease the visual quality drastically. Hence, the concealment of such block losses is a realistic situation in many cases, except for the damage of some header information.

The main motivation of this research is to demonstrate the advantages resulting from insertion of hidden data, which is related to the content of a block, into its spatio-temporal neighbors, so that this data can be used in the receiver for better error concealment. The goal of this study is not to propose a technique that will provide better error concealment quality than the approaches like mature FEC, but rather to achieve a bit stream which has an extra functionality by using data hiding for error concealment.

1.2 Outline Of Dissertation

Chapter 2 Basics on data hiding techniques are given and some of its applications are explained.

Chapter 3 Related work on error concealment techniques is presented briefly.

Chapter 4 Error concealment method based on data hiding is proposed for digital video transmission.

Chapter 5 Experimental results of the proposed method and conventional methods are given for different test sequences in different bit rates and different channel conditions.

Chapter 6 Concluding remarks are discussed along with future work for possible improvements.

CHAPTER 2

DATA HIDING AND ITS APPLICATIONS

The word “steganography”, general name for data hiding (information hiding) techniques, is literally known as “covered writing”, which comes from Greek. It is the art and science of secret communication and based on hiding information in other information.

People have tried to hide information for various purposes, since the archaic periods. Famous examples of steganography go back to antiquity. According to a story from Herodotus, a slave’s head was shaved by his master, Histiæus, and tattooed with a secret message around 440 B.C. After growing the hair back, the message disappeared and then the slave journeyed to carry the message. When he shaved his head upon arriving, the message was revealed. In another story from Herodotus, Demeratus removed the wax from a writing tablet, wrote the message on the wood and covered the tablet with wax again in order to warn the King of Persia of an attack. Some more recent steganography examples are changing the spaces between the words in a formatted text, using invisible ink, or placing imperceptible echo in some parts of an audio. More examples about steganography can be found in [7,8]. In all these examples, the main idea is embedding information into a media in a way that the cover media looks like original after embedding.

Steganography is a technique of making imperceptible modifications on the media [7,8] of any kind such as text, audio, image, and video. The growth in digital multimedia technology has made information hiding problem popular. A general scheme of steganography is given in Fig. 2.1. Any kind of data is embedded into any kind of media by a steganography technique. The data embedded media is then transmitted to a recipient via some channel. During transmission, the media may

encounter some attacks, or affect from some noise. The situation in which the hidden data is required to survive determines the type of steganography technique.

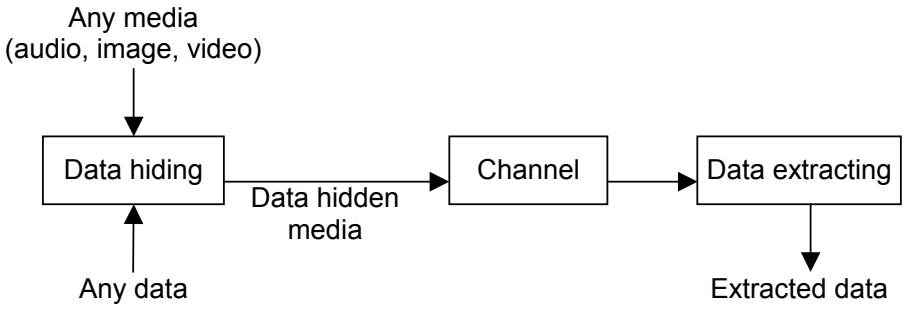


Figure 2.1. General scheme of steganography

Although cryptography, a widely known technique in data security systems, might be considered as rival to steganography, they are not competing but rather complementing branches. Information can be hidden into the encrypted media, or encrypted information can be hidden into a media. In cryptography, anyone receiving the encrypted signal realizes the secret communication, but since data hiding techniques are imperceptible, the existence of a covert communication is unknown to the other recipients in steganography. On the other hand, while cryptography protects the content of the signal, steganography conceals the existence of the data hidden in the signal.

Steganographic techniques have four fundamental constraints: imperceptibility, capacity, robustness, and security. It is not possible to satisfy all these constraints together in one technique. However, according to the requirements of any specific application, some kind of trade-off is established to develop a satisfactory steganographic technique.

Imperceptibility, or known as fidelity, of a technique is referred to the perceptual difference between the marked and original signal and can be tested subjectively [7]. The owner of an image, for example, would not accept any visual degradation on the image due to data hiding. Therefore, the hidden data should cause minimum distortion on the cover signal.

Capacity is another desired property in a steganographic system. Maximum possible amount of data should be embedded into the cover media. However, increasing the amount of data embedded in a cover signal causes the hidden mark visible and degrades the visual quality of the cover signal [8]. Therefore, imperceptibility should also be considered when the capacity of the system is tested.

Robustness of the steganography technique is desired when the marked signal will be passed through some signal processing operations (filtering, compression, etc.), or some geometrical distortions (rotation, translation, scaling, etc.). If the hidden data is still detectable after these operations and distortions, then the system is accepted to be robust. However, it is not likely for a technique to survive all of the operations. Robustness to which type of distortion strongly depends on the application.

Security of a steganography technique is referred as the resistance to the hostile attacks. Detecting the hidden information in the cover signal should not be possible even if the data hiding technique is known [7]. These attacks try to remove hidden data from a cover signal. The other types of attacks include detecting the existent hidden mark or embed another mark to the cover media. It should be noted that these attacks are designed to keep the visual quality of the cover signal in an acceptable level. However, similar to the case in the robustness, security criteria is also application dependent.

Depending on the criteria reviewed above and the application areas, a classification of the steganographic techniques can be achieved as in Fig. 2.2. Steganography can be divided into three groups: data hiding, copyright marking, (semi) fragile watermarking. Data hiding is mainly used for error concealment by transporting some error concealment data from encoder to decoder. It is not particularly robust but a secure steganography technique, which provides protection against detection.

The type of steganography signatures used for authenticating a digital content is called as fragile watermarking [8]. It protects the content against forgery by alerting any distortion on it. Copyright marking is robust to the removal attacks for hidden mark and can be divided into two subgroups [7]: watermarking and fingerprinting. On the other hand, watermarking embeds the mark of the originator into the original work, in fingerprinting for each of the recipients, a different mark is hidden in order to track the transition of the material. Moreover, watermarking can

also be classified in two groups as private and public watermarking. Private watermarking that is mostly used for broadcast monitoring and copy/play control is robust to the hostile attacks and only a group of people with a key is allowed detecting the watermark [8]. Public watermarks are not robust to the attacks and make the images “smart” as embedding a copyright notice into the images.

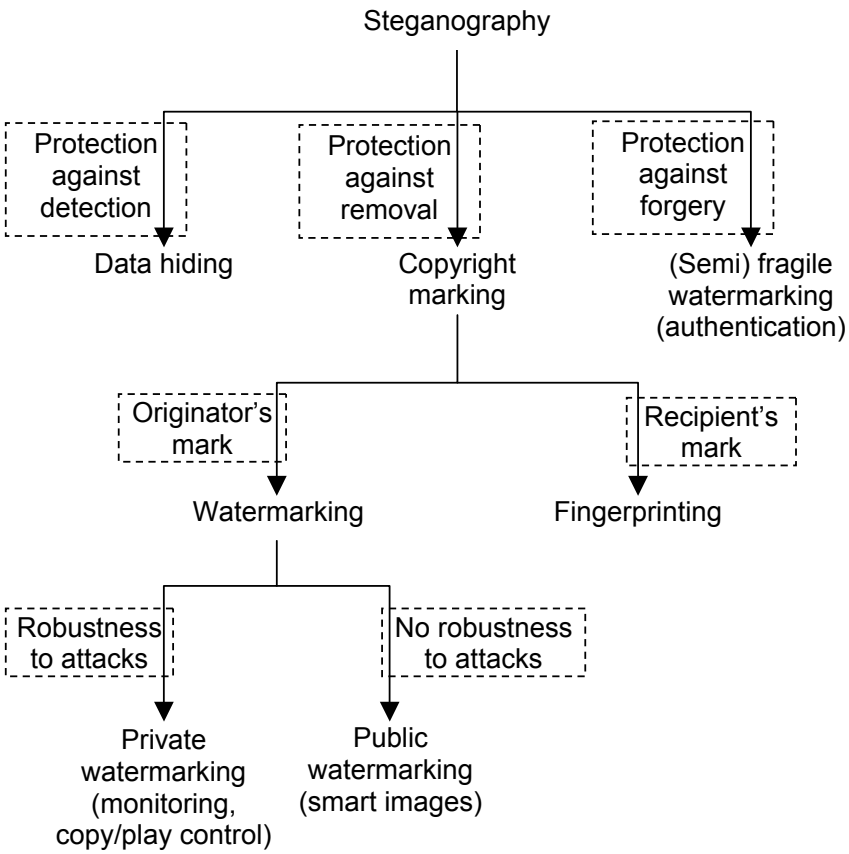


Figure 2.2. Types of steganography techniques

2.1 Applications

Since steganography is a perceptually transparent process and can be applied to any multimedia signal, it has attracted many researchers rapidly for different reasons. Today there are several data hiding applications that are either

proposed or in use such as covert communication, broadcast monitoring, copyright protection, transition tracking, content authentication, copy control, and also error concealment [7,8,10-12,16-18,31-35].

Steganography can be used as a means of covert communication. A secret message is embedded transparently into an image or video by any steganographic technique and the image or video is sent to the recipient. Unintended receivers even do not realize the communication [10,11].

Advertisers or musicians hire people to have them watch TV or listen radio in order to determine the broadcasting instants of their video clips. An automated system, which records the broadcasting time of the advertisements and video clips, is desirable for the advertisers and musicians. This so-called broadcast monitoring system can be realized by embedding some data into the advertisement or music [8]. Then a computer can monitor all the broadcast and search for the hidden data in the advertisement to record its broadcast time.

When an original Work, such as a painting, a photograph, a song, or an image, is created, a copyright notice is placed on it to identify its owner. This technology for identifying the owner of a Work does not protect the copyright holder always, since the copyright notice can easily be removed or sometimes can be neglected during the copying. In addition, copyright notices in the images may look ugly aesthetically. A steganographic system can be used for owner identification of an original Work [7]. Since embedded data is an imperceptible and degradation in visual quality is very small, data hiding is an alternative solution to the textual copyright notices.

With steganography, the leakage of a photo to the press by whom can be tracked. In order to prevent redistributing the copy of the Work illegally, different marks are embedded into each legal copy. Therefore, the owner could track the transition of the Work and find out the responsible person for misuse [8].

Digital multimedia technologies allow editing and copying any part of the digital content easily and perfectly. Undetectable modifications on the digital data have resulted with the problem of content authentication. A preferable solution is to embed a signature into the original data. In case of even the slightest modification, the hidden signature is corrupted and the system can detect that original content is not authentic [12].

Steganography can also be used instead of encryption employed in satellite television broadcasts. In many TV broadcasts the signal is encrypted with a key and the customers can watch the broadcast only with a decoder using this key. However, if someone who has a legal encryption key record and redistribute the broadcast, the cryptographic protection will be useless. If the decoders are designed to detect the marks hidden in the signal and allow watching the TV according to the existence of hidden marks, then illegal broadcasting can be prevented [8].

Recently, error resilient video transmission has become a new application area for steganography, as some novel concealment methods are proposed [16-18, 31-34]. Basically, useful data for error concealment is hidden at the encoder and transmitted to the decoder. These methods are examined in the next chapter.

2.2 Basic Steganography Techniques

Steganography techniques take different names according to the applications and the properties as depicted in Fig. 2.2. Most of them are based on substitution of the redundant, or insignificant parts of the signal. Some basic steganography techniques can be listed as low-bit modulation [8], spread spectrum modulation [10], quantization index modulation [38], and statistical methods [7].

The least significant bit (LSB) plane of the pixel values of an image is substituted with the message bits for low-bit modulation. The receiver extracts the hidden message bits if he knows which pixels are modified. Since the image is distorted very slightly in this process, the embedding capacity is high. However, the hidden data is vulnerable to the attacks. Even small corruptions on the image due to signal processing operations can distort the embedded data. Some variants of this technique include randomizing the order of the message bits to be hidden or dividing the image into regions and embedding one bit into one region.

Spread spectrum techniques embed the mark in a transform domain, such as Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), or Discrete Wavelet Transform (DWT). Modifying the transform coefficients provides more robustness to the compression, cropping, or some image processing, than LSB substitution, which is required for watermarking.

As an example, a spread spectrum technique [10] proposes to embed the watermark into the frequency domain coefficients of an image such that some

coefficients are changed by adding some pseudo random numbers to them between 0 and 1, as illustrated in Fig. 2.3. Visually significant coefficients are selected for modification, which makes the watermark more robust to the attacks while the watermarked image is still perceptually same with the original. The watermark embedded by this technique survives scaling, JPEG coding distortion, dithering distortion, cropping, print-scan process, successive watermarking attack, and collusion attack [10].

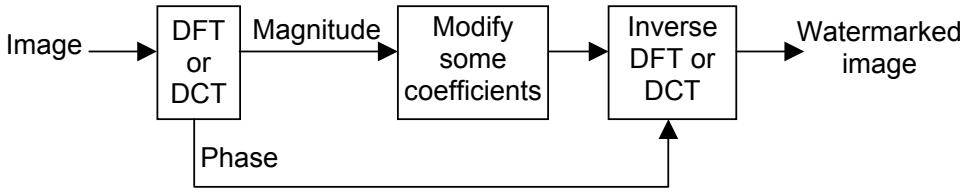


Figure 2.3. Spread spectrum watermarking

A trade-off exists between the robustness and imperceptibility in the transform domain techniques. If the amount of the hidden data increases, the watermark becomes more robust but it turns to be visible on the other hand. In order to embed data as much as possible without decreasing the imperceptibility, Human Visual System (HVS) based watermarking methods are developed [35]. Watermark is embedded into the DWT coefficients of an image according to the thresholds determined by some psychophysical tests and considering the foveation phenomena of HVS.

In another watermarking technique [11], Fourier-Mellin transform is applied to the image and the watermark is embedded in this domain as shown in Fig. 2.4 in order to provide a rotation, scaling, and translation (RST) invariant watermarking [11]. DFT magnitudes are not affected from translation so DFT domain provides robustness to translation. First, translation effect is eliminated by taking DFT magnitude and then Cartesian coordinate is converted to Log-polar coordinate in order to change the effect of rotation and scale to translation and lastly, by taking again DFT magnitude, all these effects will be eliminated. Therefore, the watermark at this final domain becomes robust to RST.

Most of the compression standards quantize the source data for better coding efficiency. This quantization step can be used for embedding data. Depending on the value of the data to be hidden, different quantizers are used in quantizing the source data. In the receiver side the hidden data can be extracted by determining which quantizer is used. One of the most popular approach which utilizes this idea is Quantization Index Modulation (QIM) [38].

In statistical steganography, some statistical characters of the cover image are changed in order to embed one bit. This technique lacks of capacity. But the image can be divided into blocks and for each block; statistical properties are modified in a different way to increase the capacity.

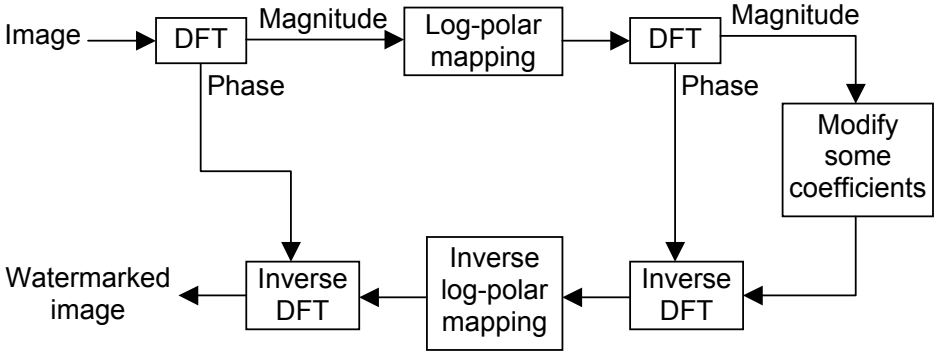


Figure 2.4. Rotation, scaling, and translation invariant watermarking

In this thesis, “even-odd” signaling of the DCT coefficients is used to hide data. The LSB of DCT coefficients are substitute with the data bits to be embedded. Since data should be inserted in the compressed domain for this kind of applications, the selected method is optimal in the sense that it does not require any robustness and gives the best performance in terms of capacity [9]. However, if the transmission standard is not predetermined, this information must be embedded into the image intensities in a more robust manner. In that case, the amount of bits to hide will decrease, considerably, and hence, total number of bits required for the method may become a critical factor.

CHAPTER 3

DECODER SIDE ERROR CONCEALMENT TECHNIQUES

Although, error concealment techniques can be divided into three major categories in the literature, as re-sending, post-processing and adding redundancy, in this thesis, we only focus on error concealment at the decoder, i.e. post-processing operations to recover lost information at the receiver side.

Most popular coding standards (MPEG, ITU-T H.263+, etc.) compress the frames of digital video in two ways, intra and inter. Intra-coding is very similar to the still image coding, e.g. JPEG, which uses the spatial correlation between the pixel values of the image. In inter-coding, temporal correlation between the video frames is utilized, besides the spatial correlation. Hence, the past research on error concealment via post processing at the decoder should be examined in two different classes, as intra- and inter-frame recovery.

Almost all error concealment techniques try to recover the damaged parts of the image or video in block-based, since widely used coding standards are block-based and the possible bit errors usually cause block errors or a slice of block errors.

3.1 Intra-frame Error Concealment

The intensities of a natural image, from one pixel to its neighborhoods, vary smoothly over the entire image, except for the regions with sharp edges. The

techniques, which reconstruct the missing blocks of still images or intra-frames of video sequences, are based on the smoothness property within an image and use the available spatially adjacent data. Most basic and popular technique is known as bilinear interpolation in spatial domain. Each pixel in the damaged block is interpolated using four nearest border pixels in four directions [1]. The interpolation weights are calculated according to the distances between the damaged pixel and correctly received pixel. The smaller the distance is the greater the weight is as depicted in Fig. 3.1 (a).

Interpolation in frequency domain is also possible in a similar way. Each DCT coefficient of the damaged block is interpolated from the corresponding coefficients of the four neighboring blocks [1]. The interpolation weights in this method are all equal, i.e. the neighborhood block coefficients that are in the same position with the damaged coefficient are averaged as illustrated in Fig. 3.1 (b).

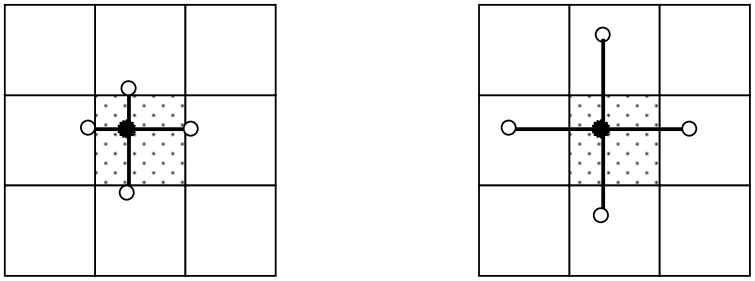


Figure 3.1. Interpolation of damaged block from its neighborhood in (a) spatial and, (b) frequency domains.

Since some parts of the image contain high variations, simple bilinear interpolation may not give satisfactory results in all cases. Blocks can have visually complex geometrical structures. Some techniques try to estimate the local geometric structure around a missing block. In [14,15], some spatial directional interpolation schemes are proposed for better interpolation. The pixel is reconstructed via interpolation of two nearest pixels from the correctly received neighborhood blocks in a predicted direction. Two nearest surrounding pixel layers of a missing block are

used to estimate an edge passing through the missing block in [14] and interpolation is performed along this edge direction as shown in Fig.3.2. Again the interpolation weights are inversely proportional to the distances of damaged and existing pixel. In a similar approach [15], the Hough transform is utilized to determine the best direction for interpolation. In these approaches, it is assumed that the visual structure around the missing block is also present in it, not exactly but in a similar way.

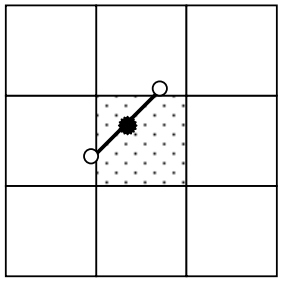


Figure 3.2. Interpolation along a direction in spatial domain

3.1.1 Utilization of data hiding for error concealment

The previous methods try to estimate the lost block from the available correctly received data. The main idea behind the approaches in this class is to insert some hidden data related to a block into its neighbors, so that this information can be used in the receiver for a better concealment. This provides a forehand data for reconstructing the damaged block and increases the reconstruction quality.

In [14], the edge of a missing block is tried to be estimated for interpolation along the edge direction. Following this approach, the major edge direction of each block is determined at the encoder and hidden into a neighboring block [16]. Therefore, it is not required to estimate any possible edge for the lost block at the decoder, since the edge information of the lost block is transported to the decoder by the help of a hidden channel within the image. Obviously, this prior information about edge direction results with more satisfying reconstruction quality.

Instead of interpolating the frequency domain coefficients of the damaged block at the receiver side; in a similar approach to hiding edge direction, the coefficients of a low quality version of each block can also be embedded into another companion block [17]. However, this approach requires considerable amount of bits to hide the coefficients for an acceptable recovery, even a coarse quantization is applied to the coefficients to be hidden. Thus, the visual quality of the video is reduced considerably due to data hiding prior to transmission, while trying to reconstruct the lost blocks by the hidden coefficients after passing of the video through the noisy channel.

Loss of synchronization is another major problem in case of bit-errors in noisy channels. The decoder may not realize the error and continue decoding the bit stream wrongly in which case the error propagates to the next blocks. Hiding a resynchronization data, such as the bit length of a block into a neighboring block, using any data hiding method can be another approach for the error concealment problem [18]. Decoding more or less bits for a block is prevented by this way.

In all of these approaches [16-18] reviewed above, the data are usually embedded into the image by *even-odd signaling* [12] of the DCT coefficients.

3.2 Inter-frame Error Concealment

In case of video transmission, the recovery of the lost blocks is a simpler problem compared to image transmission, its spatial counterpart; since temporal correlation can also be utilized in addition to the spatial correlation. The techniques in this category utilize temporal, as well as spatial smoothness property of the video together for obtaining better results. The higher temporal smoothness between video frames leads to a better concealment.

Inter-frame coding is based on finding motion information of each block between the frames and coding a motion vector (MV) for each block. Mainly, the error concealment approaches in this category try to recover the motion information of a damaged block from its neighboring blocks. In order to estimate the lost MVs, there are several major approaches [1,3], such as zero value assumption for the lost MV (i.e. replacing the damaged block with the same positioned block in the previously decoded frame), or assigning the MV of the corresponding block at the same location in the previous frame for the lost MV, or using the average/median of

the MVs from spatially adjacent blocks as an estimate for the lost MV. Similarly, in [19], the average of all the optical flow vectors in the boundary pixels of the lost block is calculated to predict the motion information of the damaged block.

As it is seen from the above methods, there may be more than one candidate to replace a lost MV. In that case, some kind of selection mechanism is needed for the best reconstruction. The boundary matching algorithm (BMA) [20] may provide a measure to select a MV from a set of candidate MVs, such as the ones mentioned in the previous paragraph or the neighboring MVs. In BMA method [20], a MV is chosen which results with minimum side match distortion, as illustrated in Fig.3.3. This distortion is determined as the sum of absolute (or squared) differences between the intensities of the immediate neighbors across the boundaries of the concealed block and its neighborhood blocks [20].

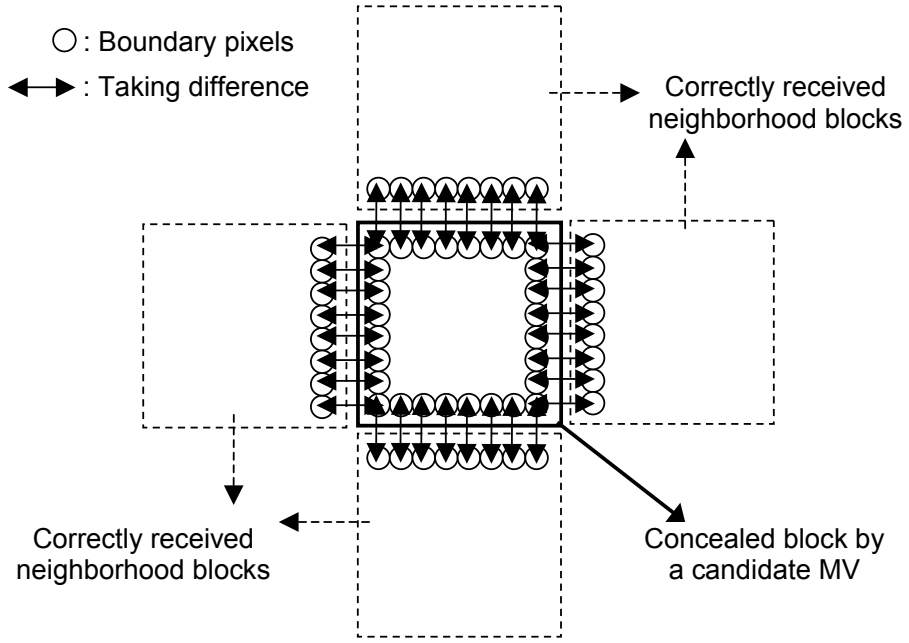


Figure 3.3. Calculation of side match distortion in BMA

BMA is utilized for reconstructing the missing block in various forms. In [21], BMA itself is used for replacing the lost block with the best matching pattern by searching in the previous frame. Then, mesh based warping is applied in order to

further fit the block with surrounding area and to reduce the artifacts caused by fast movements, rotations or deformations. As a different approach to reduce the blocky artifacts due to reconstruction, a multi-frame BMA is proposed in [22]. In this approach, boundary smoothness property of the natural images is not only considered in the current frame, but also in the succeeding motion-compensated frames. This approach is based on the fact that the neighborhood of a block is changed as the block moves from one frame to the next frame, therefore the boundary variation of a lost block should also minimized in the next frame.

In BMA techniques mentioned above, the corruption of more than one block successively in a slice is ill-considered. For especially preventing the blocky effects while concealing the slice errors, an iterative error concealment algorithm, based on BMA is proposed in [23]. In first stage, conventional BMA is applied to the successively damaged blocks in one slice. Since there is no any neighborhood to the left of the blocks in first stage, the reconstruction may not be satisfactory. Then, BMA is applied again, but for this case, the right neighborhoods of the damaged blocks are full with the previously reconstructed blocks from first stage. The iteration is stopped until total boundary errors are minimum [23].

Another approach considering the slice errors is proposed in [24]. It uses the correctly received upper and lower neighborhoods only to reconstruct the missing block. The upper and lower regions of the damaged block are searched in the previous frame to find the best matching blocks. First, the upper half of the damaged block is reconstructed after a search in the previous frame for a similar pattern of the upper neighborhood region of the damaged block. Later, the lower half of the damaged block is reconstructed similarly, yet considering also the upper half reconstructed in the first step.

Instead of using a block for searching the best match in the previous frame, the pixels of two lines are chosen around the lost block in [25] and they are used to perform the search in an area in the previous frame. The algorithm applies some weights during the search according to the type of available neighborhood data, which can be lost, concealed, or correctly received.

Another major approach for estimating the MV of a lost block is to use motion field interpolation (MFI) in which motion vector is determined for all points (not as a single motion vector for each block) in the lost block. The lost MV is

obtained by interpolation from the motion information available at a number of surrounding nodal or control points [26-30].

For each pixel of the damaged block, a candidate motion vector is found by using bilinear MFI utilizing four neighboring MVs. Another candidate MV for the damaged block is found by using BMA on the other hand. These two sets of vectors resulting from each method are combined by either averaging [26,27] or weighting [28] to assign a MV for each pixel in the missing block. Finally, the missing block is recovered by these MVs.

If the MFI technique for error concealment is applied to multi-reference codecs, which utilizes more than one reference frame for motion estimation and compensation, then, at most four candidate reconstructions can be obtained from the corresponding four neighboring MVs. Considering multi-reference codecs, an error concealment technique is proposed in [29]. The four candidates are either chosen by BMA or combined as weighted averaging.

Videos do not always contain translational motions. Complex motions, such as rotation, magnification, or reduction, should also be considered while estimating the actual motion [30]. An affine transform, which is a transformation of coordinates, can be used to model these complex motions. In [30], an affine transform is applied in order to estimate the motion parameters of the lost block using correctly received neighboring block data.

3.2.1 Utilization of data hiding for error concealment

Hiding imperceptible information for better error concealment can also be extended to inter-frames of the compressed video. The inter-coded frames can be modified to become more robust by the help of the hidden data. Since MV is very important in inter-frame coding, generally MV information is hidden for error concealment of inter-frames. There are a limited number of methods in the literature, which are tailored for inter-frame error concealment.

In [31], picture header (PH) and MVs of a frame are protected by some parity bits. Coded MVs of each block are arranged row by row as in Fig.3.4. Afterwards, they are modulo-2 summed and the resultant parity bits are hidden into the next frame. The data is embedded by modifying the motion vectors of the next frame by half-pixel, i.e. the data is embedded into MVs.

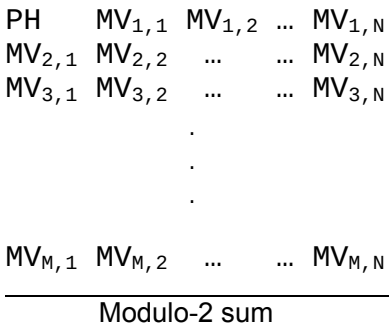


Figure 3.4. Obtaining the parity bits for inter-frame error concealment

Considering the packet losses, i.e. burst errors, the above approach is extended to a frame-wise in [32] by calculating the parity bits frame by frame for a group of frames and hiding them into the frames following the group. However, in this approach, the data is embedded into the motion compensated DCT coefficients. In addition, a block-shuffling scheme is introduced to isolate erroneous blocks [32], such that correctly received blocks surround damaged blocks.

Along with these error concealment algorithms, some error detection schemes based on data hiding are also proposed for video transmission. The method in [33], hides the parity check codes of the MBs of one frame into both the MVs and residual DCT coefficients of the next frame. Recently, a novel algorithm modifies DCT coefficients for detecting errors in the bit-stream [34]. In this method, the coefficients in a location with even index in the zigzag reordering are forced to be even, and vice versa.

CHAPTER 4

PROPOSED SYSTEM FOR ROBUST VIDEO TRANSMISSION

A novel video coding system, which utilizes data hiding to conceal transmission errors, is proposed. In the proposed system, intra- and inter-coded frames are considered separately from the error concealment point of view. For concealing the errors in intra-coded frames, mainly edge direction data of an MB and coded MB bit length value are used. On the other hand, the coded motion vector bits are utilized to conceal the errors for inter-coded frames. All these data are embedded into the video at the encoder and then extracted at the decoder as an auxiliary data for error concealment.

In the proposed system, data hiding is achieved by simple “even-odd” signaling of the DCT coefficients [12]. In order to hide “0” to a coefficient, it is forced to be even, and for “1” it is forced to be odd. Data is embedded into the LSB plane of the DCT coefficients in this way. Only the nonzero coefficients are modified, so that the run length coding rate does not increase. If all the LSBs of the nonzero coefficients are allocated for data hiding and still there are data to embed, then higher LSB planes (second, third, or fourth) are employed. If four LSB planes are not sufficient to hide data then data is not embedded into that block.

In the ITU H.263+ standard, a macroblock is composed of six 8x8 blocks: four for luminance (Y) and two for chrominance components (Cb, Cr) [13]. The data is hidden into all six blocks, homogenously, starting from the last to the first (DC) coefficient.

4.1 Intra-frame Error Concealment

In order to achieve a successful error recovery, the exact location of the error, i.e. damaged block, should be detected as a first step. After detecting the damaged block, synchronization must be established back in order to prevent the propagation of the error to the other blocks. The final step is the reconstruction of the intensities for the damaged block to finalize error recovery. Therefore, the three main issues for a successful error recovery are error detection, resynchronization and reconstruction (recovery) of the damaged block. In the remaining parts of this section, the proposed system is briefly explained.

Considering the approaches in [16,18], edge direction information [16] and bit-length data [18] are both necessary to obtain error recovery of intra-coded frames, while solving all three issues. While bit-length value is strictly necessary for synchronization, edge direction information is suitable for the reconstruction of the damaged block. Finally, these two data can be used together to detect the bit-errors.

However, the reconstruction of all the damaged blocks does not always give promising results, which causes an overconcealment case. For determining such cases and deciding for the recovery of a damaged block, a 2-bit overconcealment parity, which is obtained from DCT coefficients of the block, is proposed to accompany the edge direction information. However, the hidden data is not capable of detecting all bit errors. In order to provide a full detection capability, a single parity bit is proposed to use with the synchronization data.

In order to embed edge orientation, the block is first classified as an *edge block* by applying an edge detection algorithm. For each pixel in the block, its gradient vector magnitude and gradient vector angle are calculated by using Robert gradient operator [36]. The angles of the pixels, whose gradient magnitudes are above a threshold, are quantized into 16 equally spaced directions (i.e. represented with 4 bits) and the gradient magnitudes with the same direction are summed up. The direction with largest gradient magnitude sum is selected as the final single edge direction of the whole block (Fig. 4.1). Obviously, a single message bit should also be hidden to indicate the type of the block, i.e. an edge or a smooth block. Hence, this approach requires only 5 bits per block to embed the edge direction information to the DCT coefficients of the upper MB, which is used to recover the intensities of the blocks.

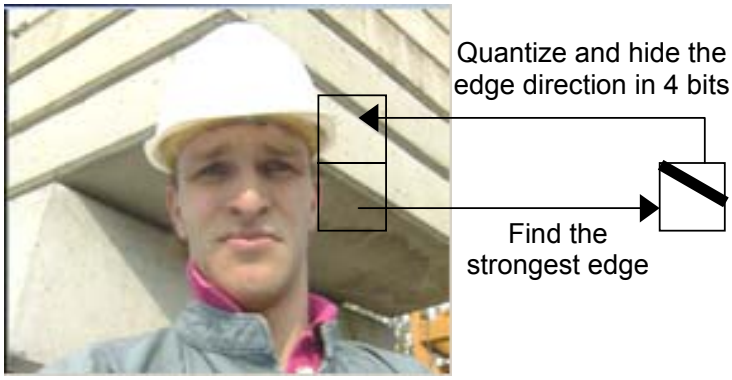


Figure 4.1. Hiding edge direction data for intra-frame error concealment

For hiding bit length data, the number of bits required for encoding the current block is determined and this value is embedded into the DCT coefficients of the previous block after being converted into binary representation during encoding (Fig. 4.2). The number of bits used in this representation should be pre-calculated by considering the maximum bit lengths of typical blocks. The proposed method requires 9 to 13 bits (i.e. for each block, bit length value varies from 511 to 8191) depending on the bit rate of the utilized video encoder (i.e. quantization parameter). By looking at the quantization parameter, decoder can determine, the number of bits used for the bit length data.

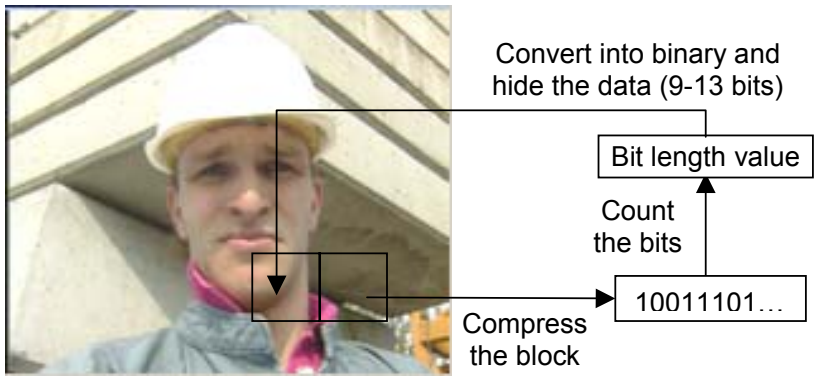


Figure 4.2. Hiding bit length value for intra-frame error concealment

4.1.1 Error detection

Both edge direction information and bit length data is used for error detection. H.263+ decoder itself can also detect errors during decompression of an MB, if it encounters a codeword that does not match with any entry in its Huffman table. If the decoder finishes decoding MB without detecting any error, then the total number of bits read from the bit-stream for decoding that block and the bit length value hidden in the previous block are compared (Fig. 4.2).

H.263+ decoder is not capable of detecting all errors. It usually does not detect an error in the codeword, and match the codeword with a wrong entry in the Huffman table. While, in some cases, few bits are decoded to reconstruct an MB, in some other case decoder propagates to the next block unwary. In order not to let the decoder propagate to the next block due to an undetected error, the hidden bit length data is checked continuously by the number of bits currently read from the bit-stream. If the bit length data embedded in previous block is not available, the decoded block's edge direction is calculated once again at the decoder and compared with the edge direction information hidden in the upper block as a secondary stage (Fig. 4.1).

4.1.1.1 Using parity

Although, checking the bit length of a block with its bit length value hidden in the previous block is enough to detect an error, it can not determine the errors, which do not change the bit length. These kinds of errors are very likely to corrupt the hidden information in that block; even though their visual damage on the block is small. This is a fundamental problem for error detection by using hidden information, which is mostly neglected in the previous methods.

In case of consistency in the bit length values, a single parity bit of the macroblock bit-stream is used in error detection. This parity bit is obtained by taking XOR of all the bits for the coded macroblock and hidden into previous block's DCT coefficients as an extra hidden information (Fig. 4.3). If the bit length value check and parity bit check do not give any error, then one can be sure that there is no error in the decoded macroblock and the data hidden in it, as well. Obviously, this is based on an assumption that there is a single bit error in the bit stream of the

current block. This assumption can be tolerable in practice, not only for binary symmetric, but also even for fading channels, if appropriate interleaving of the bit-stream can be achieved.

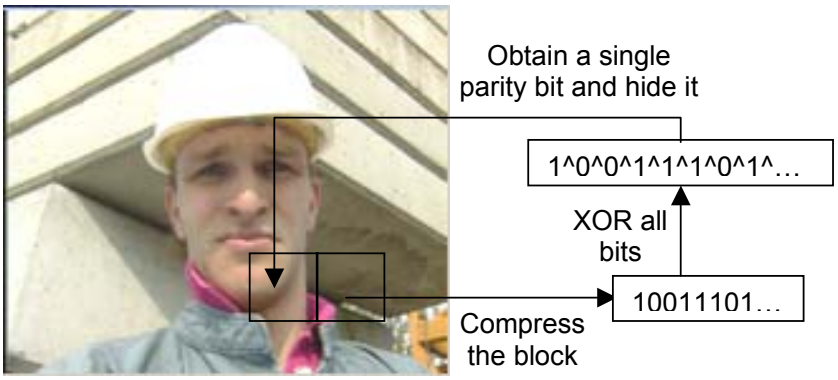


Figure 4.3. Hiding a single parity bit for intra-frame error detection

4.1.2 Resynchronization

Since the coefficients are coded in variable length, errors can easily change the bit length of the MB. This situation results with a loss of synchronization in the decoded bit stream. One can rearrange the decoder to continue from the starting point of the next MB by utilizing the hidden bit length info, which is extracted from the previous block.

After error detection, in order to resynchronize at the decoder, simply the bit length data is utilized. During decoding, the system is not allowed to decode bits more than the number that is dictated as the hidden value in the previous block. At the time of resynchronization, it is certain that the number of decoded bits is smaller or equal to the hidden value in the previous block. The difference between the hidden and decoded bit numbers is calculated and the decoder skips the calculated amount of bits, in order to start decoding from the next undamaged block (Fig. 4.4). In this way, without having macroblock headers that can guarantee synchronization, the system is able to synchronize itself at the start of each macroblock.

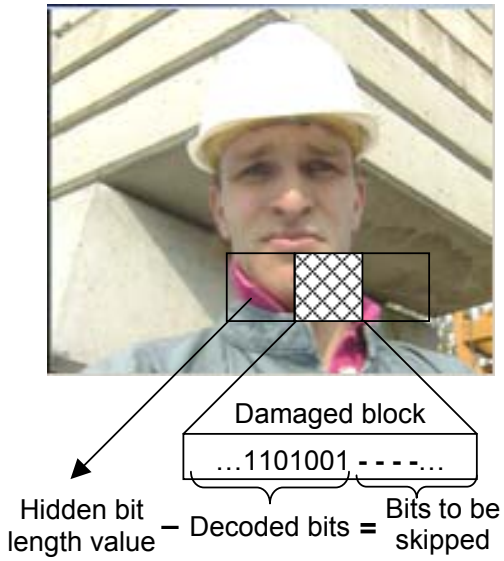


Figure 4.4. Resynchronizing the decoder to the next macroblock

4.1.3 Reconstruction

The importance of interpolation along edge direction is illustrated in Fig. 4.5. In Fig. 4.5.(a), the center block contains a strong edge with four neighboring blocks and then, this edge block is damaged in Fig. 4.5.(b). Afterwards, the damaged block is reconstructed by bilinear and edge directed interpolation techniques in (c) and (d), respectively. The results show intuitively the superiority of the edge-based interpolation, especially for the blocks with a major edge [16].

As soon as the error is detected and the synchronization is obtained, the final step is to recover the single block in which an error has occurred. For this purpose, edge direction information is extracted from the blocks in the upper slice for every block (note that the edge direction information for the blocks of the first slice is hidden into the blocks of the last slice). The first hidden bit, which indicates the type of the lost block, is tested to check whether it is an edge or a smooth block. If it is found out to be an edge block, then it is interpolated from two neighboring blocks along its edge direction (Fig. 4.6). Otherwise, for a smooth block, simple bilinear interpolation technique is applied.

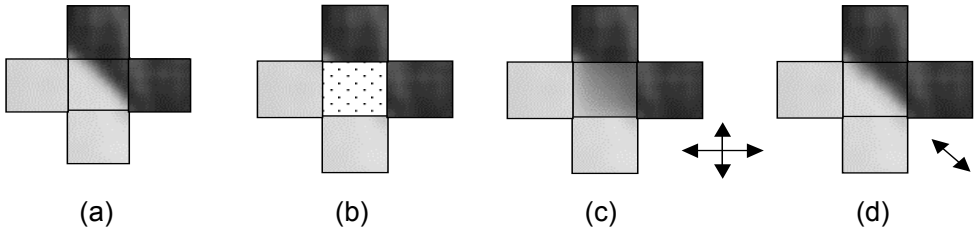


Figure 4.5. Interpolation along edge direction: (a) an edge block with four neighboring blocks, (b) damaged block, (c) result after bilinear interpolation, (d) result after interpolation along edge direction

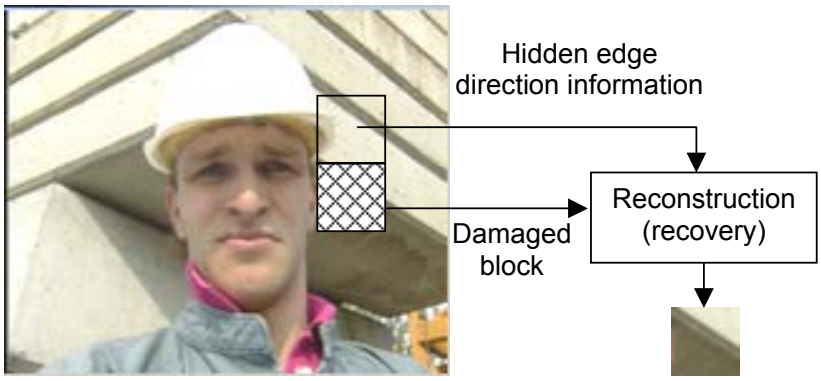


Figure 4.6. Reconstruction of the damaged block in intra-frame errors

Since the luminance component of the macroblock includes significant information about the image, edge direction based interpolation is applied only to the luminance. The chrominance components, containing smoothly varying pixels, are reconstructed by bilinear interpolation only, which still gives quite satisfactory results.

4.1.3.1 Overconcealment

After successful error detection, another important problem, neglected by previous methods, is to “measure” the visual damage at a block before recovery, since it is possible to have a small visual error in the block, undetected by the codec itself, but

“successfully” detected by the proposed system. As it is explained in Section 4.1.1.1, a parity bit is utilized to detect errors, even if they do not result with a significant visual loss.

The edge-direction based recovery technique naively tries to reconstruct the block, which has a very small visual degradation, while discarding all the available information. Such a concealment for this case is not preferable, considering the limited capability of interpolation schemes. Obviously, the reconstruction quality usually turns out to be worse than that of the available erroneous block. This situation is defined as *overconcealment* and it is avoided by using the modula-2 sum of 2 Most Significant Bits (MSB) of the current block coefficients. It is assumed that in case of visually unacceptable errors, the 2-bit MSBs are changed which can be detected by 2-bit parity hidden in the previous block and error concealment is applied for only such cases. (Fig.4.7). Note that the whole idea behind overconcealment is not to conceal, if there is not sufficient visual loss.

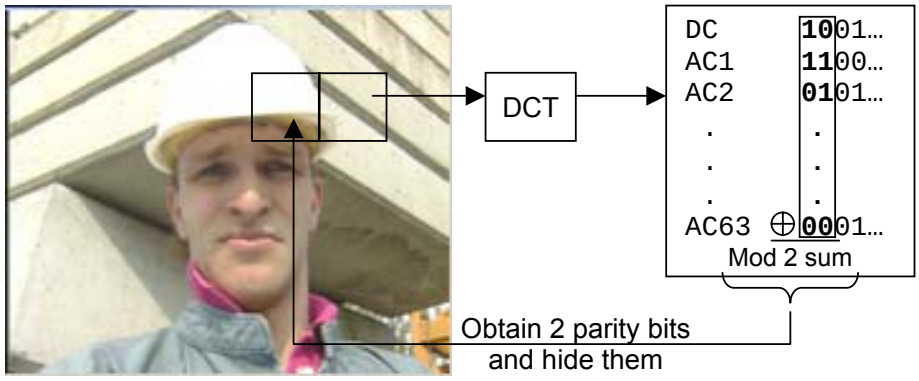


Figure 4.7. Obtaining and hiding the parity bits for overconcealment

4.1.4 Overall system

All the necessary information for intra-frame error concealment can be seen in Fig. 4.8. While the bit length, block parity, and overconcealment bits of each block are hidden into its previous block on the left of current block, the edge direction data is embedded into its upper block. All these data are concatenated and a short bit

stream is obtained. Finally, this bit stream is hidden into the neighbor block (Fig. 4.8).

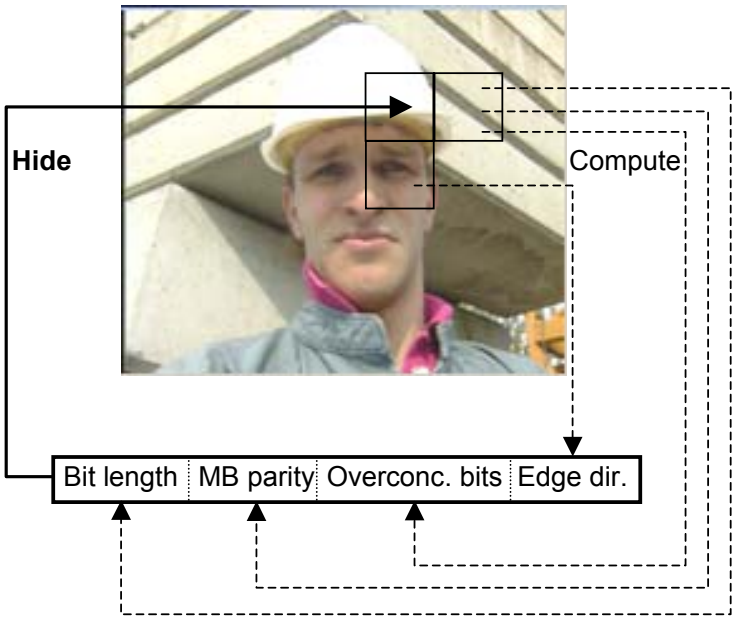


Figure 4.8. Obtaining and hiding all the necessary bits for intra-frame error concealment

4.2 Inter-frame Error Concealment

All the inter-frame error recovery methods focus on recovering the motion information of the lost block for better concealment. An obvious choice for the hidden information is motion vectors [31,32]. In addition, a checksum is utilized for detection of the errors in the hidden data.

In the proposed approach, the differential Huffman coded MV bits and coding modes (intra or inter) of the blocks in same row are concatenated and a bit stream is obtained. 9 more bits are added to the beginning of this MV bit stream for transmitting the number of bits in the bit stream to the decoder, since the MVs are coded in a variable length manner. Since error detection capability of H.263+ decoder is limited, a 5-bit checksum is also added to the end of the bit stream for the

error detection purposes in the MV bit stream. This bit stream is obtained for each row of MBs in an inter-frame and embedded into the motion compensated residual DCT coefficients of the corresponding row of MBs in the next inter-frame, as illustrated in Fig. 4.9.

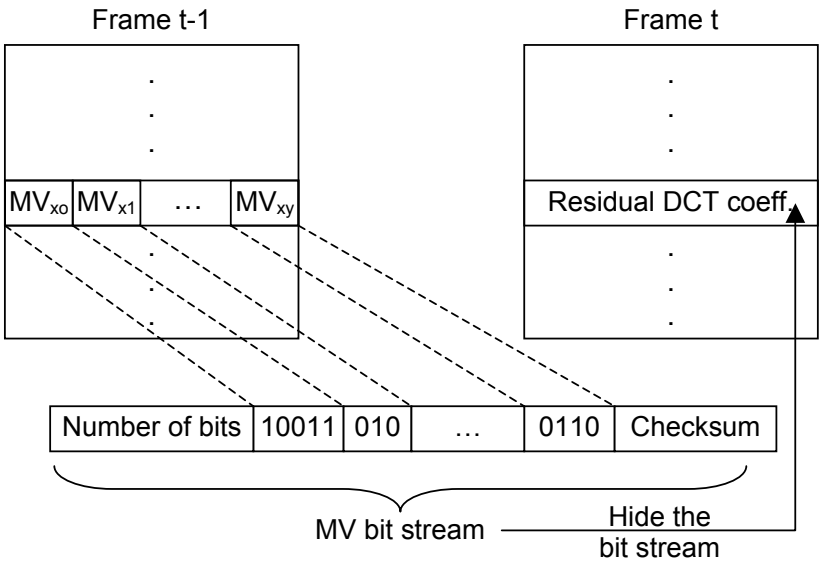


Figure 4.9. Hiding MV bits for inter-frame error concealment

Error detection step is left to H.263+ decoder for inter-frames. After the decoder detects the errors, one should wait for the next frame to be decoded in order to get the hidden MV data of the current frame. With the checksum bits, the reliability of the hidden data is verified and the damaged blocks are reconstructed by the MV information.

In some cases, only the residual DCT coefficients are affected by a bit error and decoder does not lose its synchronization, i.e. only that block is corrupted. H.263+ decoder cannot detect such errors in general. However, the data hidden in these DCT coefficients might be damaged. For these situations, a 5-bit checksum is utilized to confirm the reliability of the hidden data. In order to obtain these checksum bits, the MV bit stream is first divided into 5-bit blocks. After arranging the blocks on top of each other, they are modula-2 summed (Fig. 4.10).

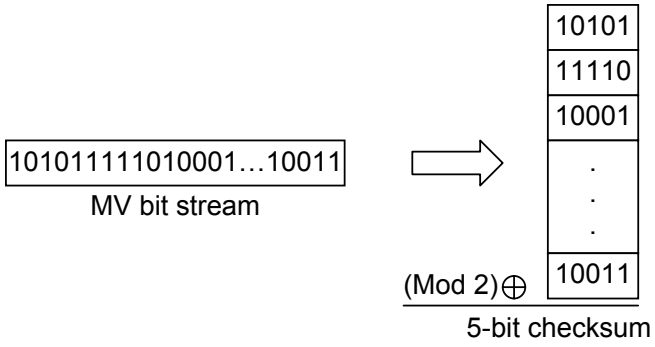


Figure 4.10. Obtaining the checksum bits

4.3 The Algorithm

An overview block diagram of the algorithm is given in Fig.4.11 and Fig. 4.12 for intra-and inter-coded frames, respectively. In both versions, there are consecutive error detection stages. The internal error detection mechanism of H.263+ is used in both inter-and intra-coded versions. For the errors invisible to the codec, the major test for intra-coded frames is synchronization and parity check, whereas the inter-coded version controls the checksum information. In addition, the intra-coded case checks overconcealment before deciding on any reconstruction.

In this overview, there are also some minor details about the proposed algorithm, such as checking continuously the reliability of the hidden data or using edge information to check errors, if the hidden data for synchronization is not available.

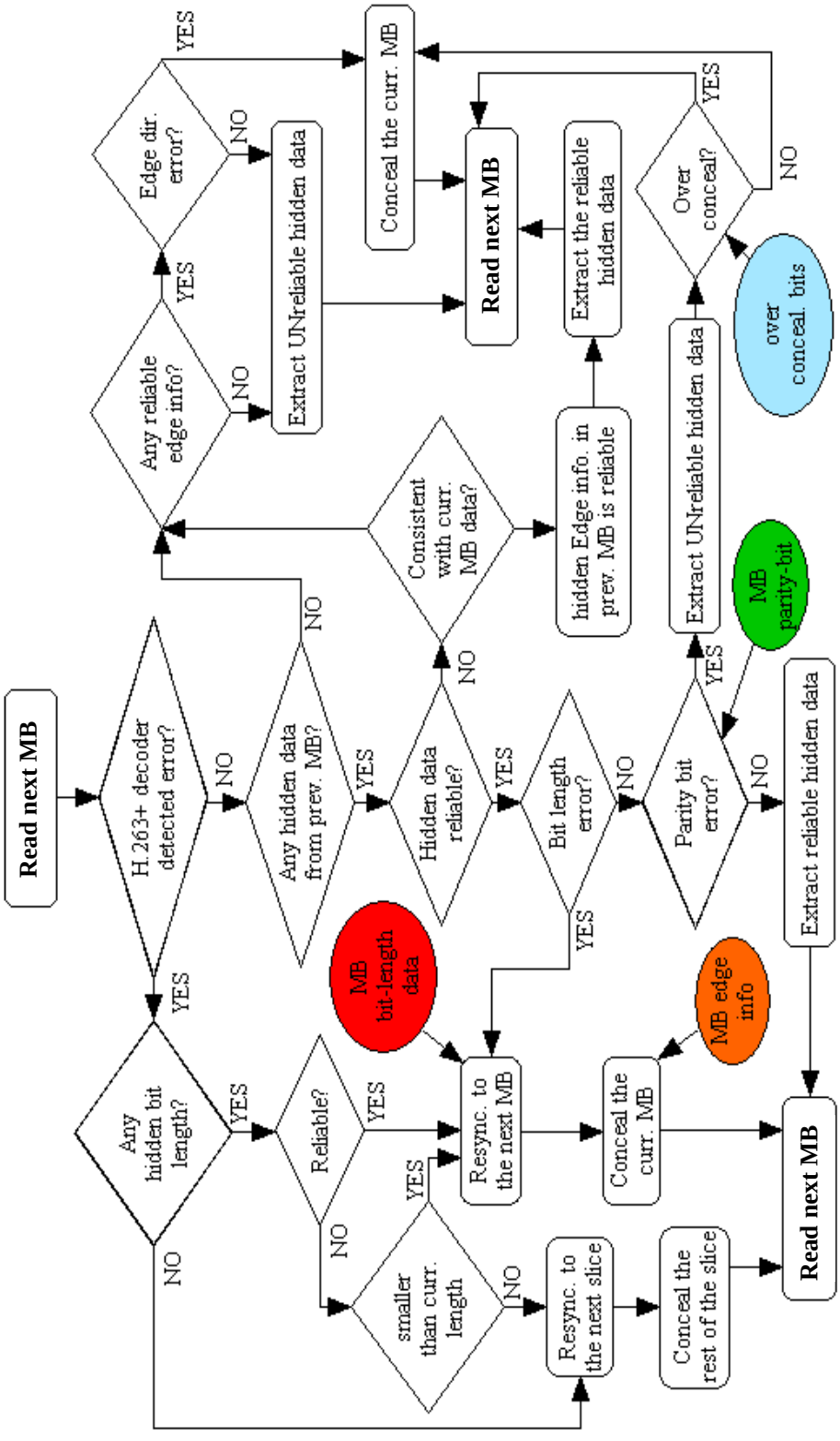


Figure 4.10. Overview block diagram of the intra frame error concealment system

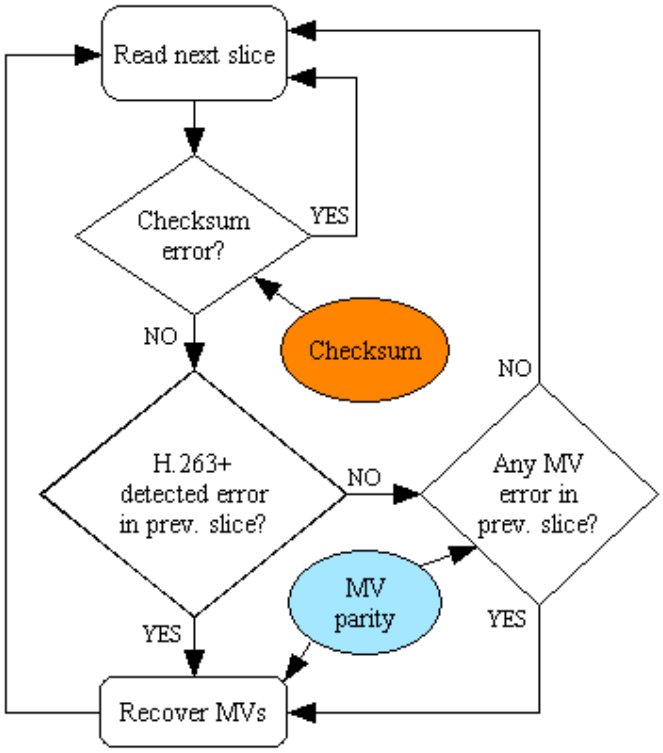


Figure 4.11. Overview block diagram of the inter-frame error concealment system

CHAPTER 5

SIMULATION RESULTS

During the experiments, a binary symmetric channel (BSC) is simulated in order to observe the effects of channel bit errors on the bit stream. For the experiments, QCIF test sequences; *Foreman*, *Carphone*, *Coast*, *Mother*, and *Table* are encoded by an ITU H.263+ codec in various bit rates and passed through the BSC for two different channel bit error rates (BER).

The visual characteristics of the test sequences show variances with respect to their motion and texture properties. The *Coast* sequence includes highly textured areas and a constant motion, which provides high frequency also in temporal domain. On the contrary, the *Mother* and *Table* sequences contain low motion and smooth regions yielding with smaller number of DCT coefficients when compared to the *Coast* sequence. However, *Foreman* and *Carphone* stand between these two types of the sequences in view of the motion and texture included in them.

5.1 Simulation Setup

Using a full encoder-decoder pair, input data is first compressed with H.263+ encoder, then this bit-stream is passed through the BSC, and finally, the corrupted bit stream is decoded using H.263+ decoder. The visual reconstruction quality is determined in terms of Peak Signal-to-Noise ratio (PSNR). This process is repeated 100 times with different random seeds for bit error pattern and average reconstructed PSNR is calculated for the luminance and chrominance components of each frame. However, in some cases, the decoder can not reconstruct the video

in the original frame number (due to the corrupted bit stream header bits), and this leads to erroneous PSNR calculations. For these situations, the video transmission simulation skips those simulations and continues with a different seed, until the original frame number is achieved at the decoder.

The above process is applied to the baseline H.263+ codec, modified codec capable of error concealment using data hiding, and finally baseline codec with some error control codes.

5.2 Performance Analysis

The proposed system is compared with the baseline decoder in these experiments. The test sequences are coded in six different bit rates and then transmitted through the BSC with two different BERs, as 10^{-4} and 10^{-5}.

The data, which will be utilized at the decoder side for error concealment, is embedded during compression of the video by ITU-T H.263+ encoder for the proposed system. The calculated PSNR values resulted from the compression and data hiding are frame wise plotted as “hidden” in the resulting figures. Afterwards, this data hidden bit stream is transmitted through the BSC and decoded by the proposed system for 100 times. The plot labeled as “concealed” shows the average PSNR values obtained by this way for each frame.

The PSNR values in the “original” plot belong to the frames of the video encoded by baseline H.263+ codec. The bit stream created by the baseline encoder is passed through the BSC and decoded by the baseline decoder again for 100 times. The average reconstructed PSNR values are labeled as “damaged”.

There is not any error concealment technique implemented at the baseline decoder. However, except for the first two frames, the blocks that could not be decoded in a frame are simply replaced by the blocks from the second previous frame at the same block location, not specifically for an error concealment.

The reconstructed PSNR value versus frame plots for luminance component only are given in the figures from Fig. 5.1 up to Fig. 5.10 for *Carphone*, *Coast*, *Foreman*, *Mother*, and *Table*, respectively. There are two figures for each video: one for BER 10^{-4} and one for BER 10^{-5}. In each figure there are six plots: one for each bit rate. In addition, average PSNR values of all frames are listed in Table 5.1 for all test sequences.

The proposed system shows better performance at high bit rates and high BER due to two reasons. Firstly, there are more coefficients available to hide data at higher bit rates, which causes a proportionally small decrease in PSNR during data hiding compared to the lower bit rates. Secondly, the small number of errors in low BER decreases the “damaged” PSNR in a small amount, even in some cases, the PSNR level for the “damaged” video is not below the “hidden” level for the BER of 10^{-5}. On the other hand, the “concealed” PSNR can not be increased to an upper level from the “hidden” level, which is already below the “original” PSNR. Therefore, the proposed system may not give satisfactory results at low BER, especially when the bit rate is also low, which is inconvenient for data hiding.

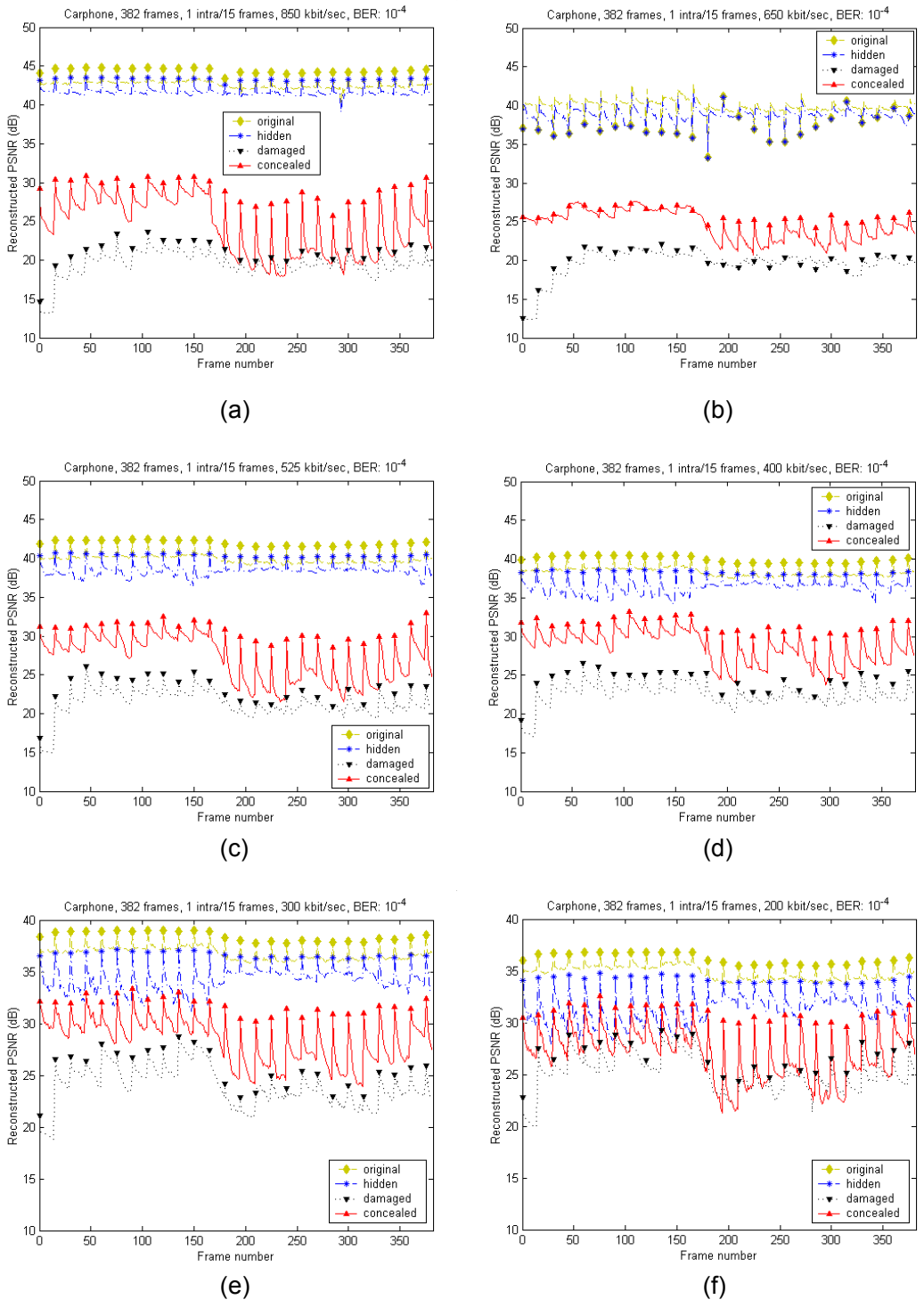


Figure 5.1. Performance comparison of the proposed system with the baseline codec for the *Carphone* sequence at the BER of 10^{-4} and at the bit rates of (a) 850 kbit/sec, (b) 650 kbit/sec, (c) 525 kbit/sec, (d) 400 kbit/sec, (e) 300 kbit/sec, (f) 200 kbit/sec.

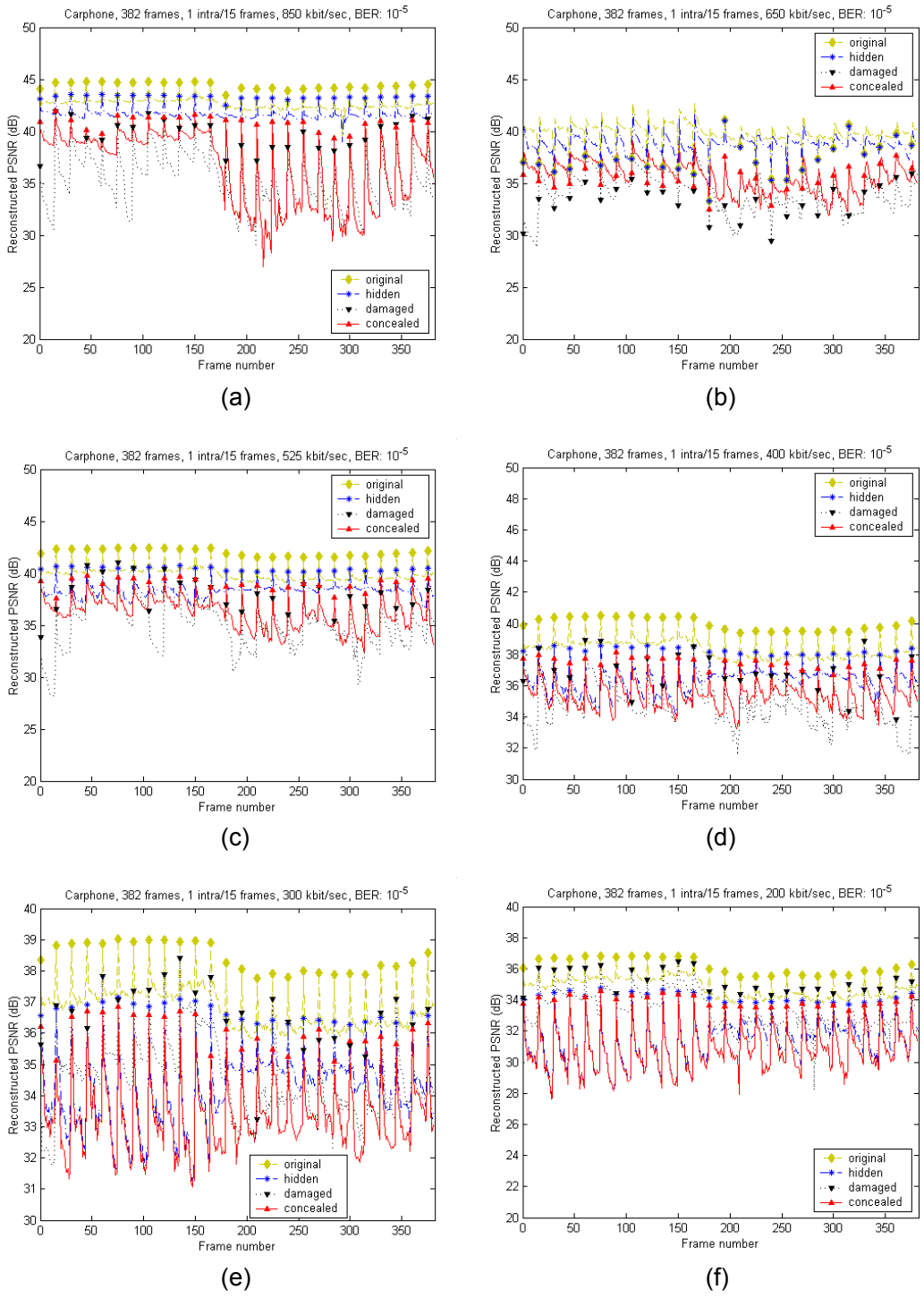


Figure 5.2. Performance comparison of the proposed system with the baseline codec for the *Carphone* sequence at the BER of 10^{-5} and at the bit rates of (a) 850 kbit/sec, (b) 650 kbit/sec, (c) 525 kbit/sec, (d) 400 kbit/sec, (e) 300 kbit/sec, (f) 200 kbit/sec.

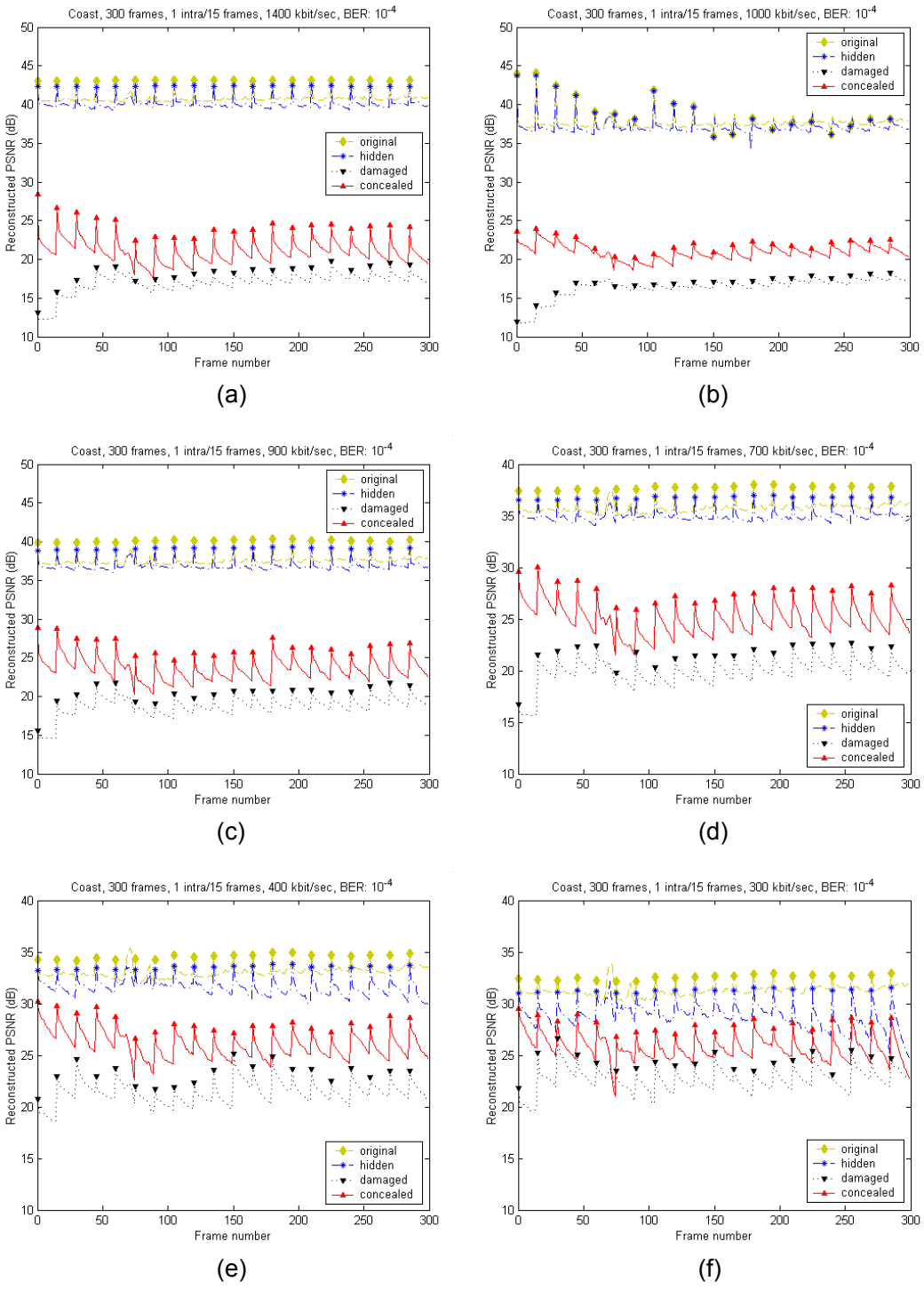


Figure 5.3. Performance comparison of the proposed system with the baseline codec for the *Coast* sequence at the BER of 10^{-4} and at the bit rates of (a) 1400 kbit/sec, (b) 1000 kbit/sec, (c) 900 kbit/sec, (d) 700 kbit/sec, (e) 400 kbit/sec, (f) 300 kbit/sec.

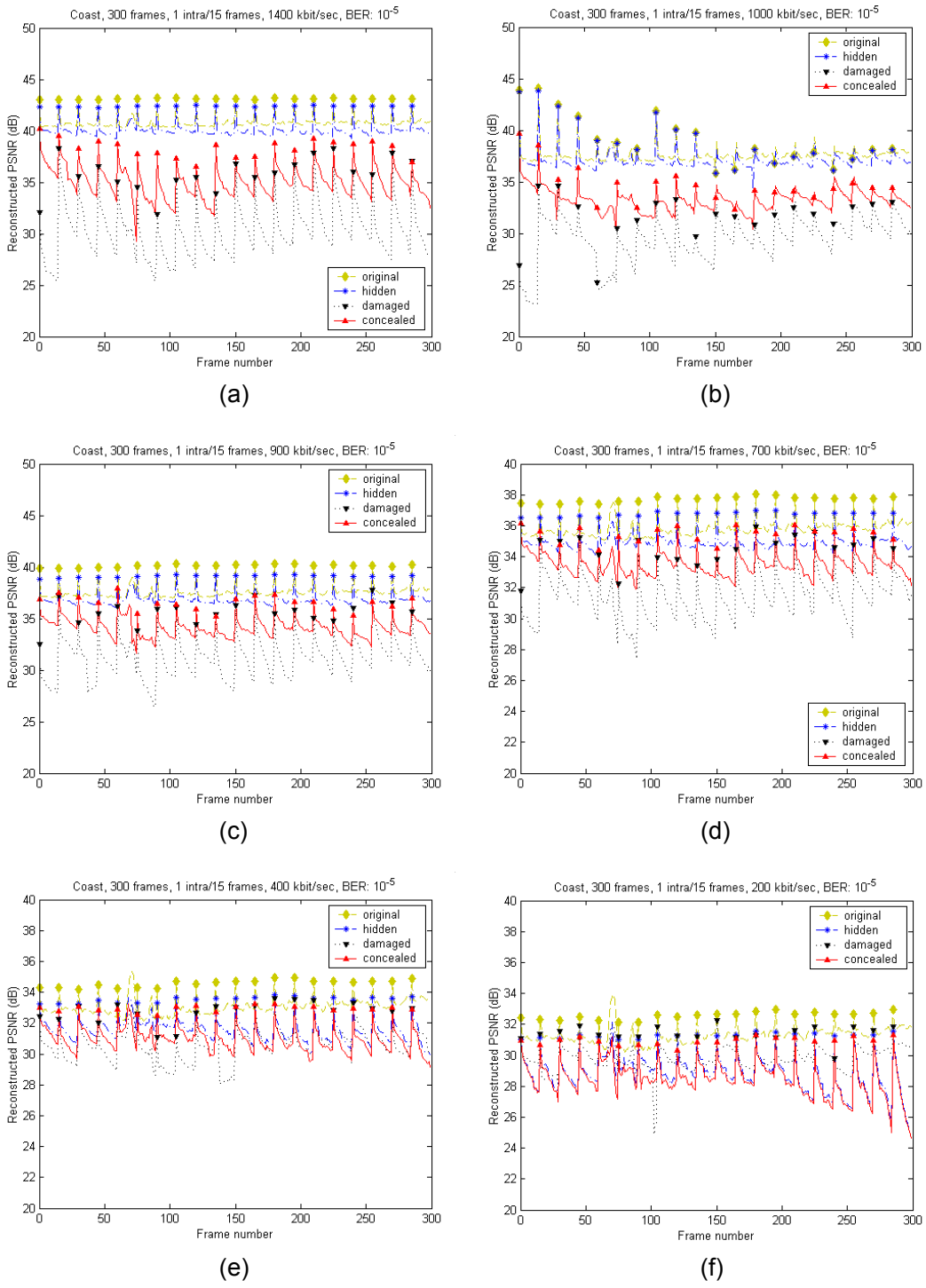


Figure 5.4. Performance comparison of the proposed system with the baseline codec for the *Coast* sequence at the BER of 10^{-5} and at the bit rates of (a) 1400 kbit/sec, (b) 1000 kbit/sec, (c) 900 kbit/sec, (d) 700 kbit/sec, (e) 400 kbit/sec, (f) 300 kbit/sec.

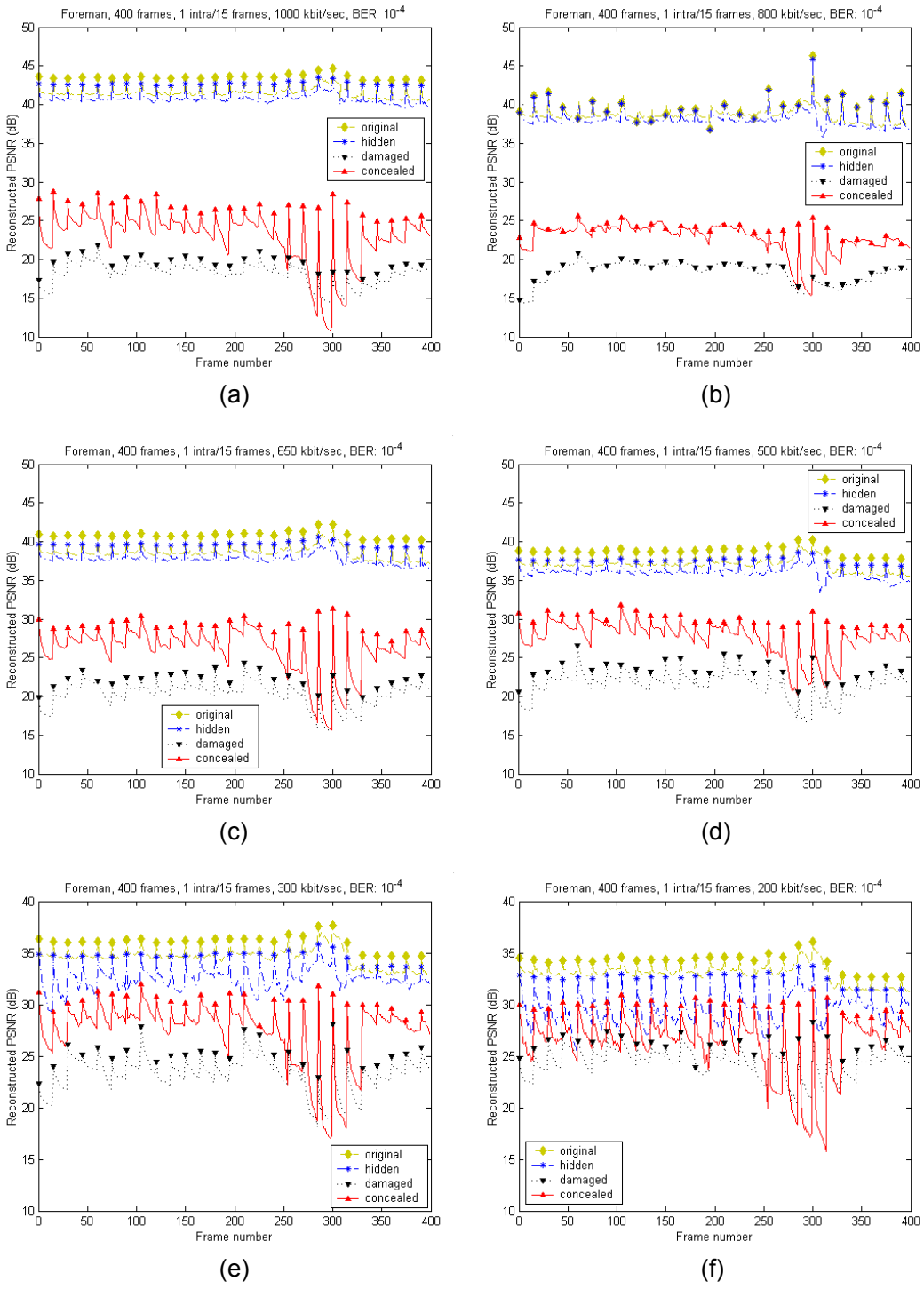


Figure 5.5. Performance comparison of the proposed system with the baseline codec for the *Foreman* sequence at the BER of 10^{-4} and at the bit rates of (a) 1000 kbit/sec, (b) 800 kbit/sec, (c) 650 kbit/sec, (d) 500 kbit/sec, (e) 300 kbit/sec, (f) 200 kbit/sec.

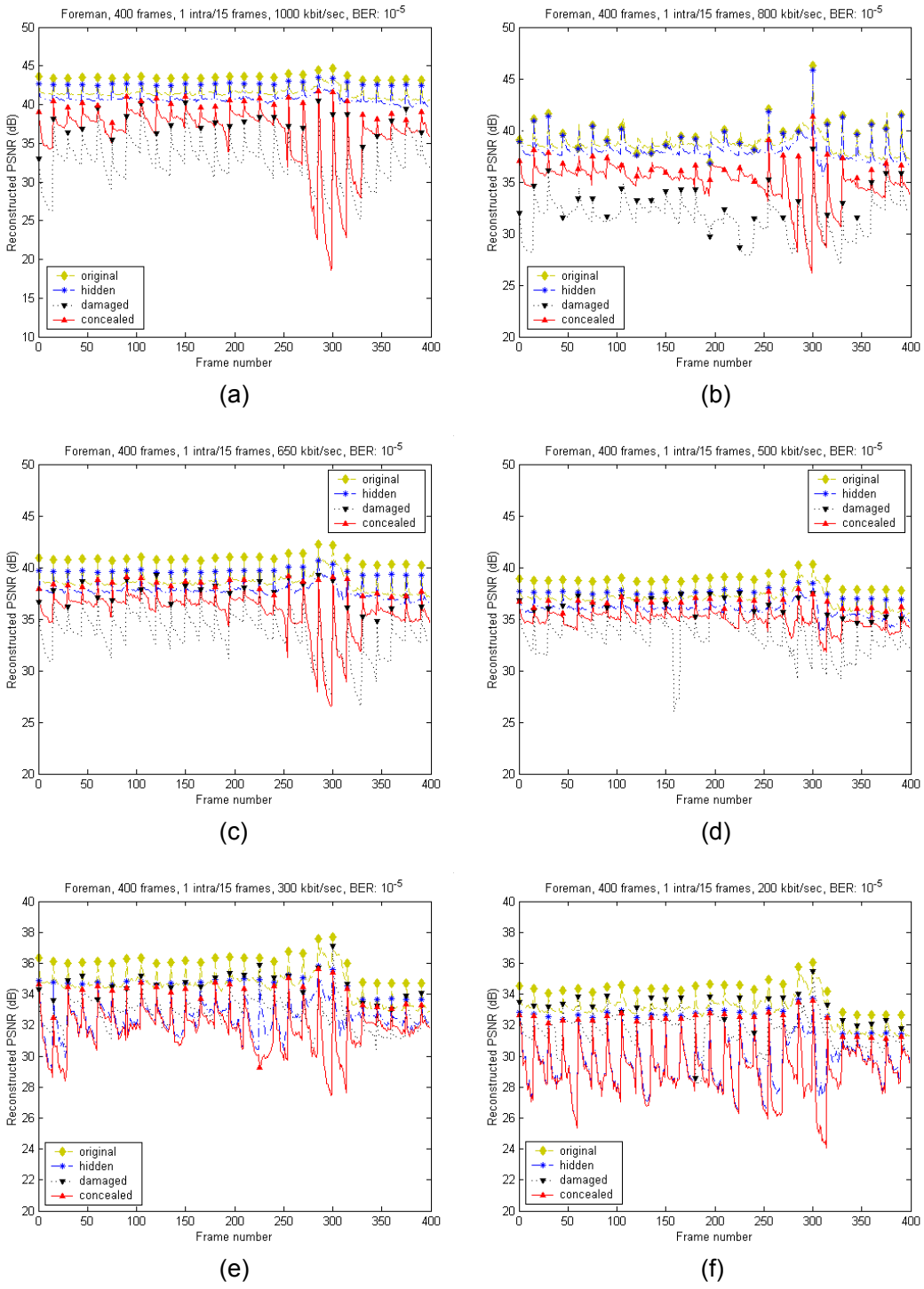


Figure 5.6. Performance comparison of the proposed system with the baseline codec for the *Foreman* sequence at the BER of 10^{-5} and at the bit rates of (a) 1000 kbit/sec, (b) 800 kbit/sec, (c) 650 kbit/sec, (d) 500 kbit/sec, (e) 300 kbit/sec, (f) 200 kbit/sec.

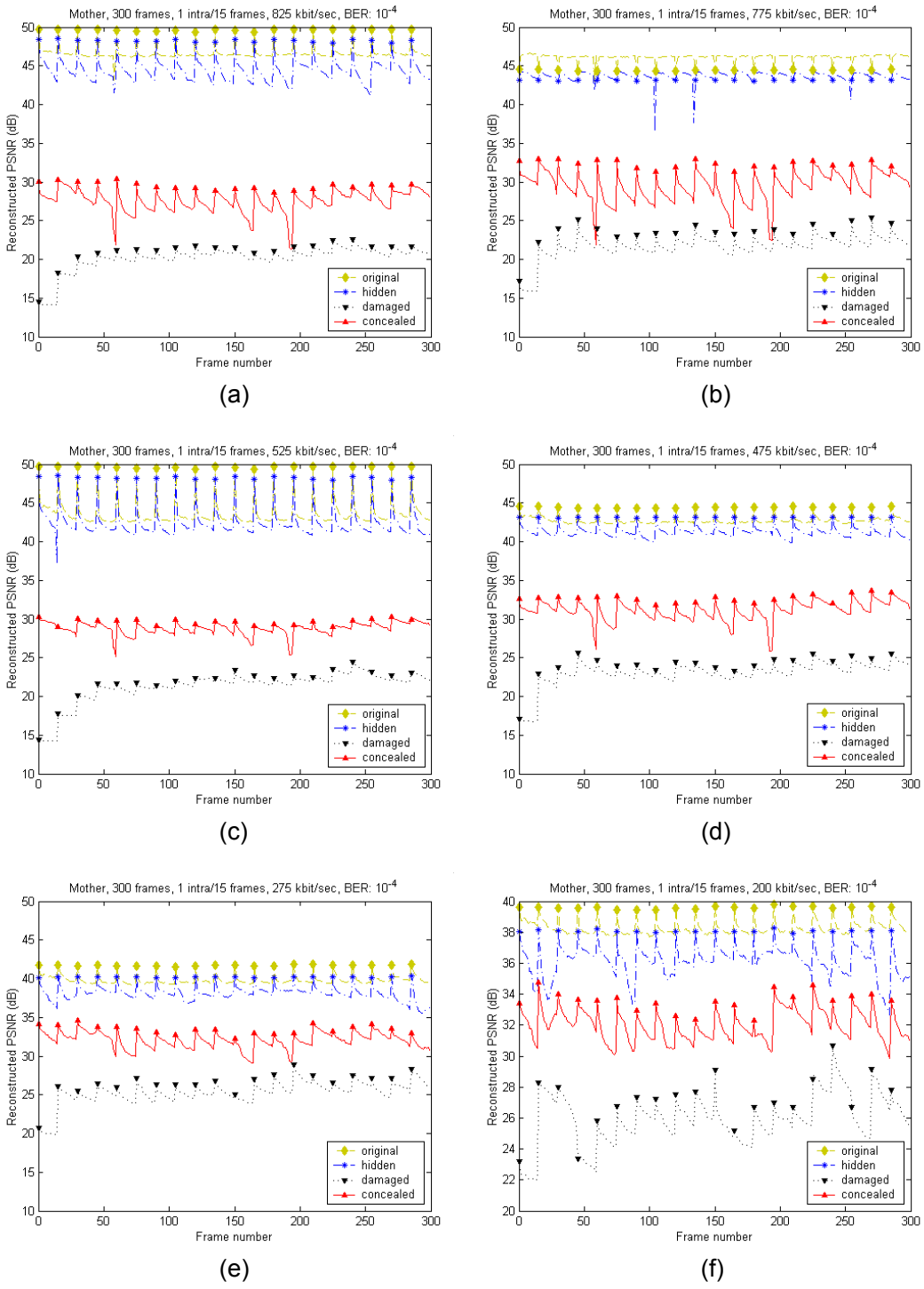


Figure 5.7. Performance comparison of the proposed system with the baseline codec for the *Mother* sequence at the BER of 10^{-4} and at the bit rates of (a) 825 kbit/sec, (b) 775 kbit/sec, (c) 525 kbit/sec, (d) 475 kbit/sec, (e) 275 kbit/sec, (f) 200 kbit/sec.

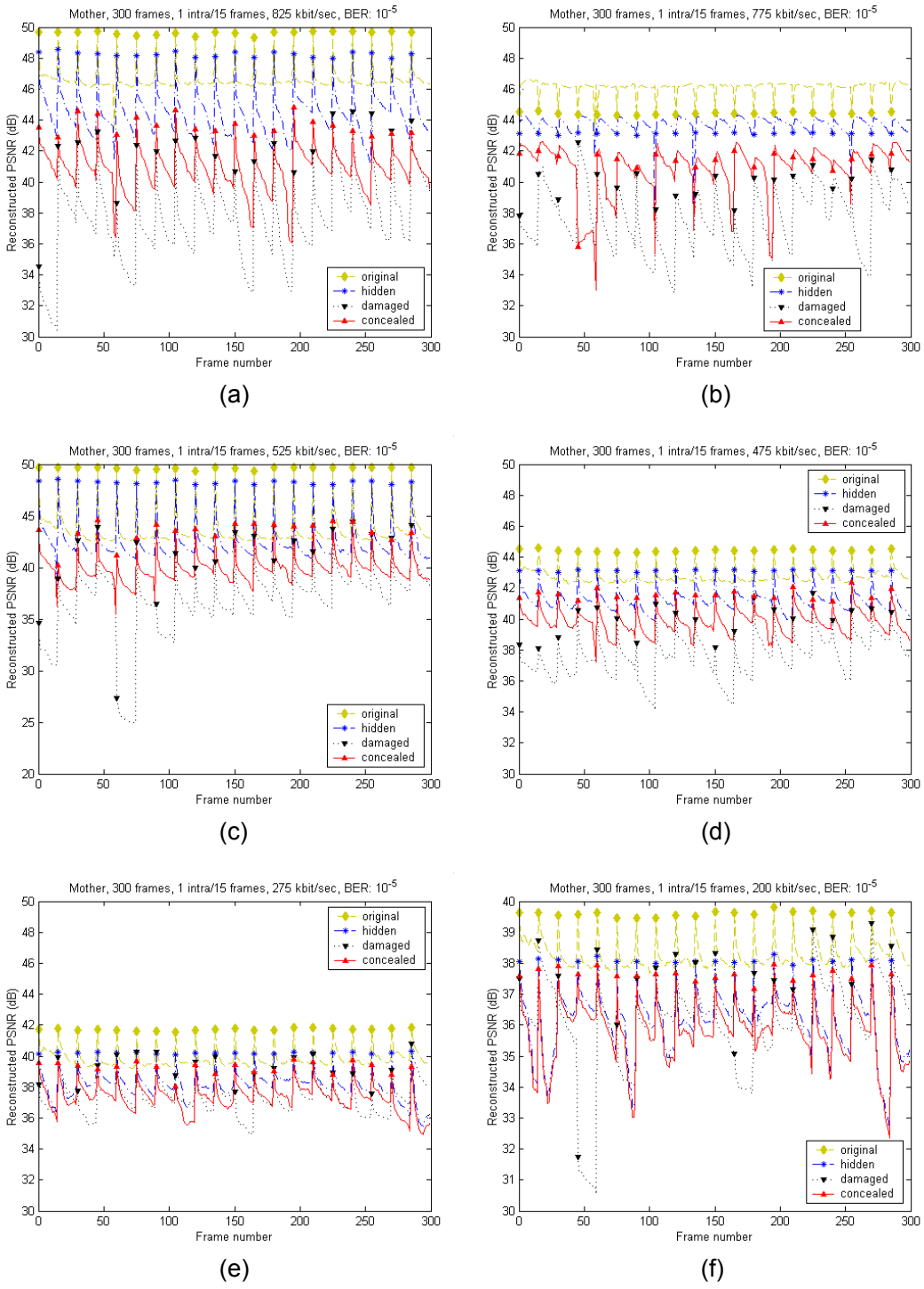


Figure 5.8. Performance comparison of the proposed system with the baseline codec for the *Mother* sequence at the BER of 10^{-5} and at the bit rates of (a) 825 kbit/sec, (b) 775 kbit/sec, (c) 525 kbit/sec, (d) 475 kbit/sec, (e) 275 kbit/sec, (f) 200 kbit/sec.

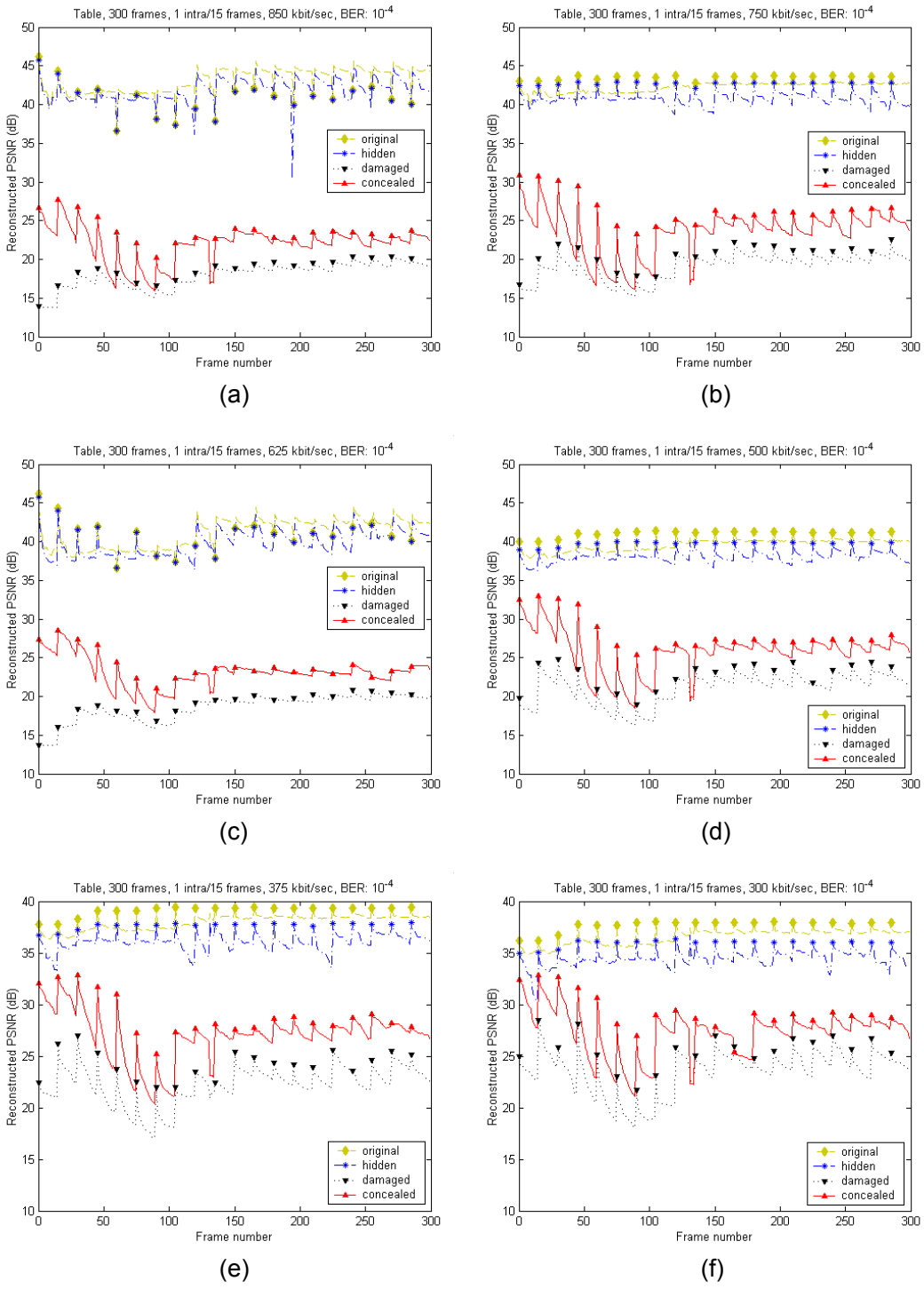


Figure 5.9. Performance comparison of the proposed system with the baseline codec for the *Table* sequence at the BER of 10^{-4} and at the bit rates of (a) 850 kbit/sec, (b) 750 kbit/sec, (c) 625 kbit/sec, (d) 500 kbit/sec, (e) 375 kbit/sec, (f) 300 kbit/sec.

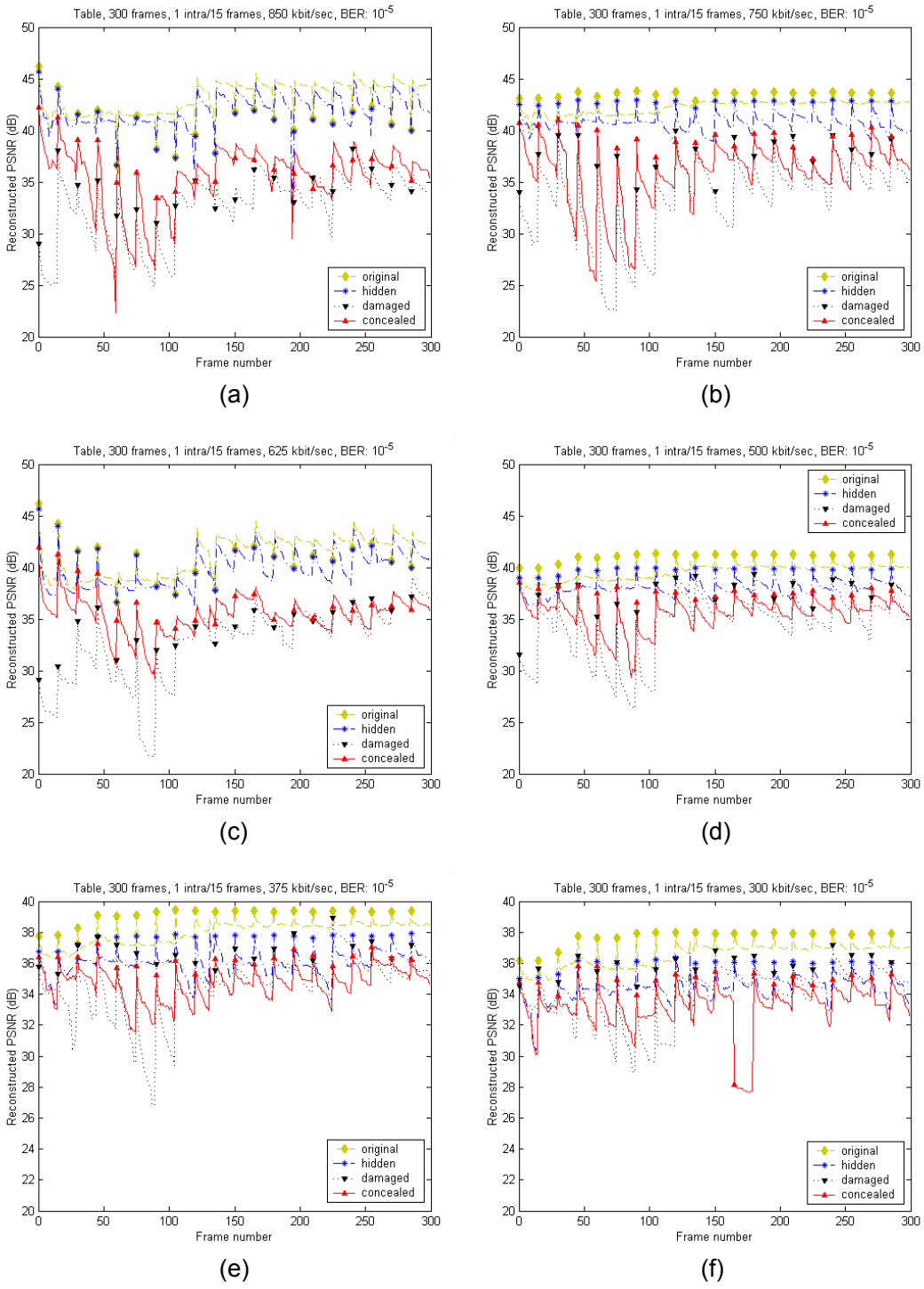


Figure 5.10. Performance comparison of the proposed system with the baseline codec for the *Table* sequence at the BER of 10^{-5} and at the bit rates of (a) 850 kbit/sec, (b) 750 kbit/sec, (c) 625 kbit/sec, (d) 500 kbit/sec, (e) 375 kbit/sec, (f) 300 kbit/sec.

Table 5.1. Average PSNR values of all frames reconstructed by the proposed system and the baseline codec under the different BERs and bit rates for the sequences (a) *Carphone*, (b) *Coast*, (c) *Foreman*, (d) *Mother*, (e) *Table*.

(a)

Carphone Average PSNR (dB)		850 kbit/sec	650 kbit/sec	525 kbit/sec	400 kbit/sec	300 kbit/sec	200 kbit/sec
No error	Original	42.69	39.91	39.98	38.38	36.83	34.86
	Hidden	41.65	38.80	38.41	36.39	34.21	31.38
10^{-4}	Damaged	19.50	19.63	21.39	22.56	23.98	25.10
	Concealed	24.49	24.60	27.16	28.67	28.44	26.99
10^{-5}	Damaged	35.25	34.01	34.86	34.72	34.57	33.45
	Concealed	36.49	35.63	36.21	35.37	33.33	30.81

(b)

Coast Average PSNR (dB)		1400 kbit/sec	1000 kbit/sec	900 kbit/sec	700 kbit/sec	400 kbit/sec	300 kbit/sec
No error	Original	40.83	37.62	37.62	35.85	33.12	31.33
	Hidden	40.07	37.05	36.89	34.96	31.57	28.81
10^{-4}	Damaged	17.00	16.42	18.77	19.87	21.41	22.65
	Concealed	21.02	20.95	23.54	24.98	26.00	25.73
10^{-5}	Damaged	30.28	29.48	31.20	31.38	30.59	29.87
	Concealed	35.04	33.27	34.16	33.50	30.98	28.49

(c)

Foreman Average PSNR (dB)		1000 kbit/sec	800 kbit/sec	650 kbit/sec	500 kbit/sec	300 kbit/sec	200 kbit/sec
No error	Original	41.56	38.73	38.66	37.09	34.64	33.02
	Hidden	40.80	38.06	37.81	36.00	32.52	29.59
10^{-4}	Damaged	18.17	18.32	20.31	21.50	23.23	24.38
	Concealed	22.87	22.74	25.97	27.60	27.29	26.15
10^{-5}	Damaged	32.62	31.49	33.57	33.31	32.74	31.66
	Concealed	36.16	34.96	35.67	34.94	31.68	29.07

(d)

Mother Average PSNR (dB)		825 kbit/sec	775 kbit/sec	525 kbit/sec	475 kbit/sec	275 kbit/sec	200 kbit/sec
No error	Original	46.65	46.11	43.66	42.78	39.87	38.28
	Hidden	44.46	43.69	42.42	41.27	38.13	36.15
10^{-4}	Damaged	20.19	21.68	21.29	23.06	25.32	25.98
	Concealed	27.59	29.43	28.83	31.14	32.08	32.02
10^{-5}	Damaged	37.57	37.64	36.73	37.63	37.28	36.09
	Concealed	40.86	40.91	39.85	39.81	37.35	35.78

(e)

Table Average PSNR (dB)		850 kbit/sec	750 kbit/sec	625 kbit/sec	500 kbit/sec	375 kbit/sec	300 kbit/sec
No error	Original	42.69	39.91	39.98	38.38	36.83	34.86
	Hidden	41.65	38.80	38.41	36.39	34.21	31.38
10^{-4}	Damaged	19.50	19.63	21.39	22.56	23.98	25.10
	Concealed	24.49	24.60	27.16	28.67	28.44	26.99
10^{-5}	Damaged	35.25	34.01	34.86	34.72	34.57	33.45
	Concealed	36.49	35.63	36.21	35.37	33.33	30.81

5.3 Comparing Main Components Of The System

The effects of the main components, which are edge direction information, bit length value, single MB parity, overconcealment bits, MV bits, and checksum, to the overall proposed system is examined during this part of the experiments. The performance of the proposed system without one of its component is tested in order to observe the absence of that component.

In the overall system without the edge direction information, reconstruction is performed by bilinear interpolation. All the other components of the overall system exist in the “w/o edge info” case. On the other hand, since 1-bit parity is utilized after the synchronization check by hidden bit length data, it is removed from the system along with the synchronization data in the “w/o synch info” case. Similarly, since overconcealment bits are designed for detecting the visually insignificant errors, they are applicable after 1-bit parity check. Therefore, overconcealment bits are extracted from the overall system for “w/o synch info” and “w/o 1-bit parity” cases, apart from the “w/o overconc.” case. Obviously, checksum bits are also not employed without hidden motion vector data.

The bit streams are coded by H.263+ encoder in which overall system is implemented as the components are extracted for each different case. Then they are passed through BSC 100 times, as in the Section 5.2 and the average luminance PSNR results for *Carphone*, *Coast*, *Foreman*, *Mother*, and *Table* are tabulated in Table 5.2 to Table 5.6, respectively.

In these simulations it is observed that the synchronization data (bit length value) is the most important component of the overall system among others, especially at higher bit rates. The absence of synchronization information decreases the reconstructed PSNR at most. Since modified H.263+ decoder knows the bit length of an MB before starting to decode it, the errors, which change the bit length and distort the synchronization, can be detected easily. Also the propagation of the errors to the next MBs is prevented by this synchronization data. These facilities of the synchronization data make it a very crucial component. When the overall system is compared with “W/o synch info” case for *Carphone* sequence in 850 kbit/sec, the PSNR loss in overall system due to hiding synch info is 0.37 dB and PSNR gain is 2.31 dB and 4.28 dB for the BERs of 10^{-5} and 10^{-4}, respectively.

Apart from synchronization data, 1-bit parity is also another important component due to its functionality about providing reliability of the hidden data. Some errors do not the change the bit length of the MB and can not be detected by synchronization data. Those errors missed by the synchronization information can be detected by 1-bit parity. Since this type of errors can destroy the hidden data also, 1-bit parity check verifies the correctness of the hidden data. Hence, the system can utilize hidden data correctly due to 1-bit parity and increases the reconstruction quality. For example, in *Foreman* sequence for 1000 kbit/sec, in spite the decrease in PSNR due to hiding 1-bit parity is 0.06 dB, the PSNR gains are 1.18 dB and 2.79 dB for the BERs of 10^{-4} and 10^{-5}, respectively.

Although the overconcealment and the edge direction components have already proved their usefulness for the tested video sequences, in some situations overall system gives lower reconstruction results against the cases without them. This is observed particularly for the *Coast* sequence. Possible reasons for this situation are the high frequency contents and fast movements included in the *Coast* sequence, which cause so crowded MSB plane that the overconcealment bits can not distinguish the real overconcealment cases. Also the edge direction based interpolation for reconstructing these busy blocks may sometimes be inadequate. However, as an example of their necessity, in the experiments for *Mother* sequence in 825 kbit/sec, while the PSNR losses for the cases “W/o edge info” and “W/o overconc.” are 0.13 dB and 0.08, respectively, the gains are 0.47 dB and 0.44 dB for the BER of 10^{-5}.

It is also observed that MV data is necessary for reconstructing the damaged inter-frame blocks, since most of the information about a block is stored in MVs. Moreover, MV data becomes more useful, if a checksum accompanies it, since the checksum provides the reliability of the hidden MV data. Checksum and MV data provide 2.25 dB and 2.20 dB PSNR gains separately for *Foreman* sequence in 1000 kbit/sec according to the overall system without them in spite of the losses of 0.03 dB and 0.42 dB, respectively. Besides, the reconstructed PSNR value of the system without checksum is sometimes less than that of without motion vector data, since hiding motion vector data requires relatively large number of coefficients and, in turn, decreases the error free reconstruction PSNR.

In error free reconstruction PSNR, the overall system is expected to have the minimum value among the others, since the amount of hidden data is maximum for

overall system. However, for some bit rates, the error free reconstructed PSNR of the system without checksum is less than that of the overall system, because data can not hidden due to insufficient number of DCT coefficients, as in Table sequence of 850 kbit/sec and 625 kbit/sec.

Table 5.2. Average luminance PSNR values of all frames reconstructed by the proposed overall system and without its components under the different BERs and bit rates for *Carphone*.

Avg. recons. PSNR Carphone (lum.)		850 kbit/sec	650 kbit/sec	525 kbit/sec	400 kbit/sec	300 kbit/sec	200 kbit/sec
10^{-4}	**Overall system**	24.49	24.60	27.16	28.67	28.44	26.99
	W/o edge info	22.90	22.96	25.53	27.29	26.98	26.27
	W/o synch info	20.21	19.76	22.23	23.47	23.56	23.72
	W/o 1-bit parity	21.53	22.05	23.22	24.51	24.75	23.97
	W/o overconc.	24.00	24.14	26.63	27.89	27.89	26.59
	W/o mot. vec.	22.32	23.06	24.46	25.97	27.29	27.84
	W/o checksum	22.81	23.24	25.40	26.45	26.78	25.41
10^{-5}	**Overall system**	36.49	35.63	36.21	35.37	33.33	30.81
	W/o edge info	35.98	35.23	36.02	35.32	33.41	30.88
	W/o synch info	34.18	32.70	34.72	34.69	32.87	30.23
	W/o 1-bit parity	35.55	34.72	35.22	34.68	32.99	30.34
	W/o overconc.	36.38	35.38	36.07	35.27	33.43	30.87
	W/o mot. vec.	35.99	35.60	36.01	35.61	34.65	33.10
	W/o checksum	34.04	34.22	34.94	34.44	32.92	30.34
Error free	**Overall system**	41.65	38.80	38.41	36.39	34.21	31.38
	W/o edge info	41.76	38.82	38.55	36.58	34.40	31.59
	W/o synch info	42.02	38.86	38.85	36.89	34.65	31.85
	W/o 1-bit parity	41.75	38.81	38.51	36.52	34.35	31.54
	W/o overconc.	41.71	38.81	38.47	36.47	34.30	31.50
	W/o mot. vec.	42.15	39.80	39.34	37.59	36.04	33.89
	W/o checksum	41.68	38.59	38.55	36.44	34.40	31.50

Table 5.3. Average luminance PSNR values of all frames reconstructed by the proposed overall system and without its components under the different BERs and bit rates for *Coast*.

Avg. recons. PSNR Coast (lum.)		1400 kbit/sec	1000 kbit/sec	900 kbit/sec	700 kbit/sec	400 kbit/sec	300 kbit/sec
10^{-4}	**Overall system**	21.02	20.95	23.54	24.98	26.00	25.73
	W/o edge info	20.67	20.63	23.07	24.36	25.94	25.60
	W/o synch info	17.91	17.44	19.35	20.16	21.61	22.33
	W/o 1-bit parity	20.30	20.80	21.87	22.77	24.04	23.81
	W/o overconc.	21.18	21.03	23.22	24.77	26.14	25.61
	W/o mot. vec.	19.67	19.81	21.42	22.59	24.18	24.89
	W/o checksum	20.57	20.32	22.32	23.45	24.51	24.70
10^{-5}	**Overall system**	35.04	33.27	34.16	33.50	30.98	28.49
	W/o edge info	34.95	33.41	34.25	33.64	31.09	28.63
	W/o synch info	30.96	28.65	31.80	31.64	30.01	28.18
	W/o 1-bit parity	34.56	33.32	33.81	32.76	30.60	28.42
	W/o overconc.	34.77	33.31	34.09	33.42	31.05	28.55
	W/o mot. vec.	31.82	32.01	32.29	32.03	30.97	29.66
	W/o checksum	32.38	31.50	32.48	32.39	30.53	28.32
Error free	**Overall system**	40.07	37.05	36.89	34.97	31.57	28.81
	W/o edge info	40.13	37.06	36.95	35.03	31.68	28.94
	W/o synch info	40.26	37.09	37.10	35.21	31.84	29.12
	W/o 1-bit parity	40.11	37.06	36.93	35.02	31.64	28.90
	W/o overconc.	40.09	37.06	36.92	35.00	31.63	28.88
	W/o mot. vec.	40.50	37.56	37.30	35.47	32.63	30.68
	W/o checksum	40.11	36.69	36.93	35.03	31.74	28.90

Table 5.4. Average luminance PSNR values of all frames reconstructed by the proposed overall system and without its components under the different BERs and bit rates for *Foreman*.

Avg. recons. PSNR Foreman (lum.)		1000 kbit/sec	800 kbit/sec	650 kbit/sec	500 kbit/sec	300 kbit/sec	200 kbit/sec
10^{-4}	**Overall system**	22.87	22.74	25.97	27.60	27.29	26.15
	W/o edge info	20.91	20.46	23.68	25.23	26.12	24.97
	W/o synch info	18.99	18.55	21.17	22.73	23.30	23.35
	W/o 1-bit parity	20.08	20.31	22.20	22.96	23.68	23.47
	W/o overconc.	22.24	22.23	25.34	26.79	26.76	25.77
	W/o mot. vec.	20.39	20.99	22.92	24.24	25.76	26.74
	W/o checksum	21.54	21.59	24.04	25.51	25.70	25.21
10^{-5}	**Overall system**	36.16	34.97	35.67	34.94	31.68	29.07
	W/o edge info	35.52	34.38	35.23	34.71	31.72	29.14
	W/o synch info	33.38	31.58	33.84	33.83	31.26	28.95
	W/o 1-bit parity	34.98	34.07	34.85	34.31	31.41	28.89
	W/o overconc.	35.92	34.77	35.72	34.91	31.68	29.11
	W/o mot. vec.	33.81	34.20	34.56	34.41	32.96	31.55
	W/o checksum	33.86	33.67	34.34	33.97	31.26	28.88
Error free	**Overall system**	40.80	38.06	37.81	36.00	32.52	29.59
	W/o edge info	40.87	38.08	37.89	36.08	32.63	29.72
	W/o synch info	41.02	38.11	38.06	36.29	32.82	29.91
	W/o 1-bit parity	40.86	38.07	37.88	36.08	32.61	29.69
	W/o overconc.	40.83	38.07	37.85	36.05	32.59	29.66
	W/o mot. vec.	41.22	38.65	38.30	36.67	34.15	32.39
	W/o checksum	40.83	37.70	37.85	36.07	32.68	29.74

Table 5.5. Average luminance PSNR values of all frames reconstructed by the proposed overall system and without its components under the different BERs and bit rates for *Mother*.

Avg. recons. PSNR Mother (lum.)		825 kbit/sec	775 kbit/sec	525 kbit/sec	475 kbit/sec	275 kbit/sec	200 kbit/sec
10^{-4}	**Overall system**	27.59	29.43	28.83	31.14	32.08	32.02
	W/o edge info	25.61	27.88	26.68	29.22	30.81	31.19
	W/o synch info	21.64	23.24	22.09	23.77	25.01	25.81
	W/o 1-bit parity	24.95	25.95	26.26	26.98	27.67	27.87
	W/o overconc.	27.42	28.96	28.63	30.81	31.93	31.93
	W/o mot. vec.	25.35	25.71	27.17	28.89	30.69	31.34
	W/o checksum	27.47	28.77	28.09	30.07	31.16	31.51
10^{-5}	**Overall system**	40.87	40.91	39.85	39.81	37.36	35.78
	W/o edge info	40.40	40.84	39.49	39.64	37.28	35.83
	W/o synch info	37.12	38.25	36.31	37.37	35.40	34.63
	W/o 1-bit parity	39.77	40.34	38.94	38.84	36.16	35.25
	W/o overconc.	40.43	40.90	39.72	39.60	37.40	35.81
	W/o mot. vec.	39.21	38.49	39.06	39.37	37.73	36.56
	W/o checksum	39.66	39.92	38.88	38.96	37.17	35.69
Error free	**Overall system**	44.47	43.69	42.42	41.27	38.13	36.15
	W/o edge info	44.60	43.99	42.51	41.43	38.28	36.33
	W/o synch info	44.96	44.87	42.74	41.89	38.70	36.75
	W/o 1-bit parity	44.59	43.92	42.50	41.40	38.24	36.27
	W/o overconc.	44.55	43.87	42.48	41.37	38.21	36.22
	W/o mot. vec.	45.79	44.28	43.20	41.86	38.95	37.25
	W/o checksum	44.51	43.50	42.05	41.21	38.29	36.31

Table 5.6. Average luminance PSNR values of all frames reconstructed by the proposed overall system and without its components under the different BERs and bit rates for *Table*.

Avg. recons. PSNR Table (lum.)		850 kbit/sec	750 kbit/sec	625 kbit/sec	500 kbit/sec	375 kbit/sec	300 kbit/sec
10^{-4}	**Overall system**	21.97	23.51	23.09	25.76	27.02	27.32
	W/o edge info	21.12	22.91	21.95	25.04	26.44	27.09
	W/o synch info	18.54	19.80	18.99	21.71	22.93	23.67
	W/o 1-bit parity	20.76	21.74	21.82	23.55	24.61	25.10
	W/o overconc.	21.95	23.23	22.85	25.60	26.78	27.12
	W/o mot. vec.	20.88	21.94	21.82	24.03	25.05	26.24
	W/o checksum	21.29	22.61	22.31	24.62	25.74	26.07
10^{-5}	**Overall system**	35.45	35.79	35.41	35.47	34.70	33.13
	W/o edge info	35.17	35.50	35.18	35.29	34.66	33.47
	W/o synch info	31.81	33.49	31.79	33.97	34.22	33.02
	W/o 1-bit parity	34.56	35.20	34.41	34.69	34.30	32.51
	W/o overconc.	35.28	35.58	35.26	35.23	34.65	33.21
	W/o mot. vec.	34.07	34.57	35.10	34.94	34.13	33.48
	W/o checksum	33.09	33.96	34.13	34.45	34.00	32.78
Error free	**Overall system**	41.88	40.77	39.83	38.05	36.11	34.44
	W/o edge info	41.91	40.84	39.85	38.18	36.29	34.70
	W/o synch info	42.03	41.11	39.92	38.60	36.77	35.22
	W/o 1-bit parity	41.91	40.83	39.85	38.17	36.31	34.63
	W/o overconc.	41.90	40.82	39.84	38.13	36.26	34.58
	W/o mot. vec.	42.87	41.76	40.97	38.83	37.04	35.41
	W/o checksum	41.04	40.77	39.52	38.11	36.15	34.45

5.4 Performance Comparison With Error Control Codes

A popular way of correcting errors is using Error Control Coding (ECC), which is widely used in digital communication systems and digital storage systems [37]. The systematic codes place some parity symbols at the end of information symbols and create a codeword as shown in Fig. 5.11. Then, by the help of parity symbols, they correct the possible bit errors on the codeword. Obviously, if the number of parity symbols used for error correction increases then the correction capability of the ECC becomes higher. However, the bit rate overhead also increases in this case.

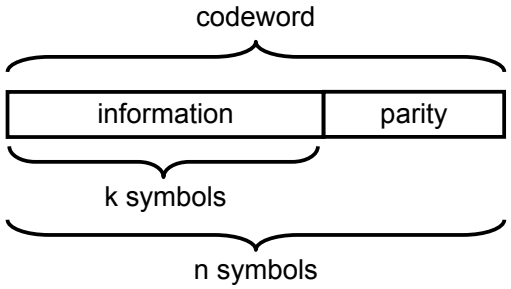


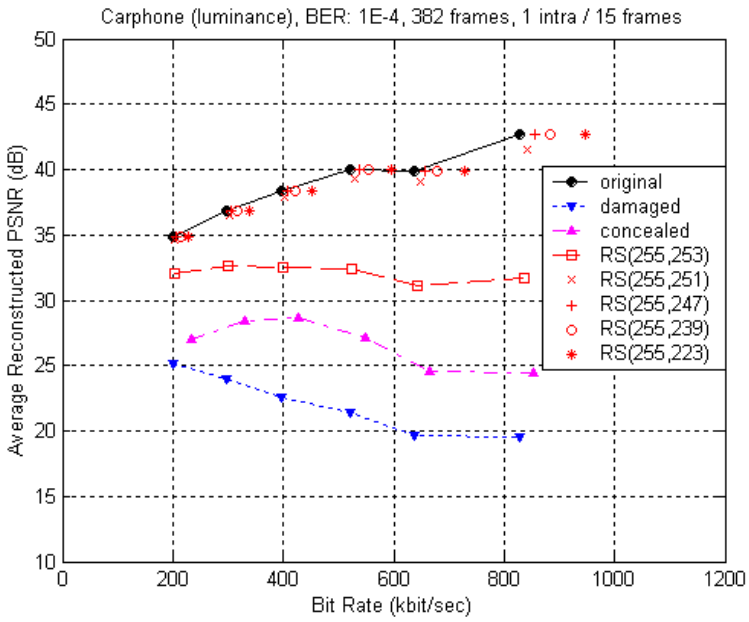
Figure 5.11. General structure of a codeword in the ECC

In these experiments, the proposed system is compared against Reed-Solomon (RS) coding, which is a well-known ECC. RS coding is chosen, since it is a powerful and widely known code in the literature. RS codes are implemented in 5 different (n, k) parameters: (255, 253), (255, 251), (255, 247), (255, 239), and (255, 223). All of these codes are in the field of 2^8 elements, Galois Field 2^8, or GF (2^8). In other words, the information and parity symbols are composed of 8 bits. Therefore, (255, 253) means that information is 253 bytes long and there is a 2-byte parity at the end of it. This RS (255,253) code can correct 1 symbol error in the codeword, since the number of symbols that an RS code can correct is equal to the half of (n-k) [37].

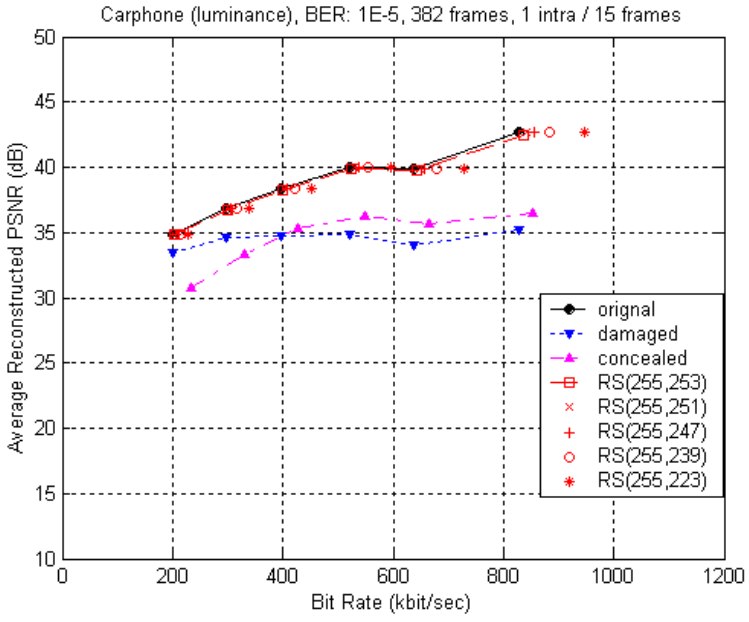
The RS codes are added to H.263+ coded bit stream and the bit stream is passed through the BSC 100 times at the BERs of 10^{-4} and 10^{-5} as in the previous sections. The source videos are coded at 6 different bit rates. PSNR values of all

reconstructed frames are averaged for each bit rate and plotted in the figures from Fig. 5.12 to Fig. 5.16 for the sequences of *Carphone*, *Coast*, *Foreman*, *Mother*, and *Table*. The plots labeled as “original”, “damaged”, and “concealed” refer to the reconstruction with no errors by the baseline codec, reconstruction with errors by the baseline codec and reconstruction with errors by the proposed system.

In these plots it is observed that although the proposed system provides higher reconstruction quality than the baseline codec, in noisy channel conditions RS codes give superior results than the proposed system. However, the advantages of the H.263+ codec as a result of the proposed system with capabilities of error detection, resynchronization, and selection the type of reconstruction, should also be taken into account in such a comparison.

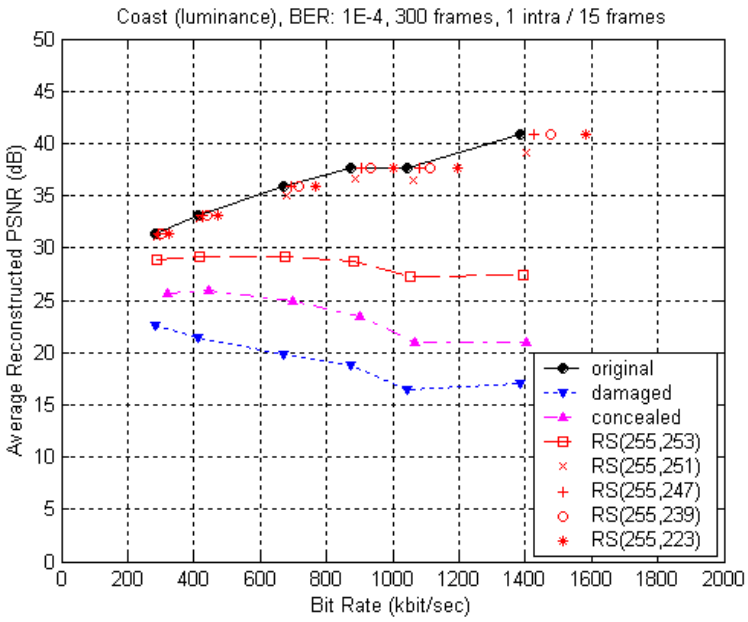


(a)

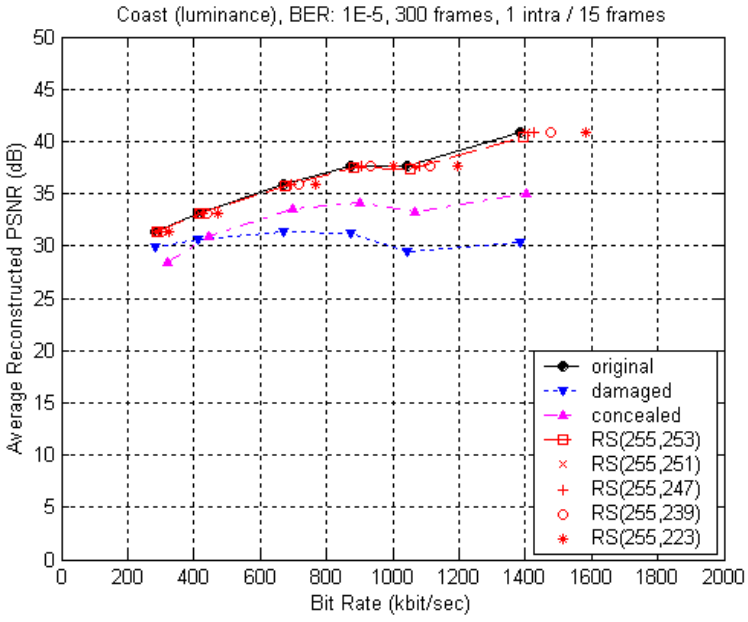


(b)

Figure 5.12. Performance comparison of the proposed system with the Reed-Solomon codes for the *Carphone* sequence: average reconstructed PSNR values of all frames vs. channel rate at the BER of (a)10^{-4} and (b)10^{-5}.

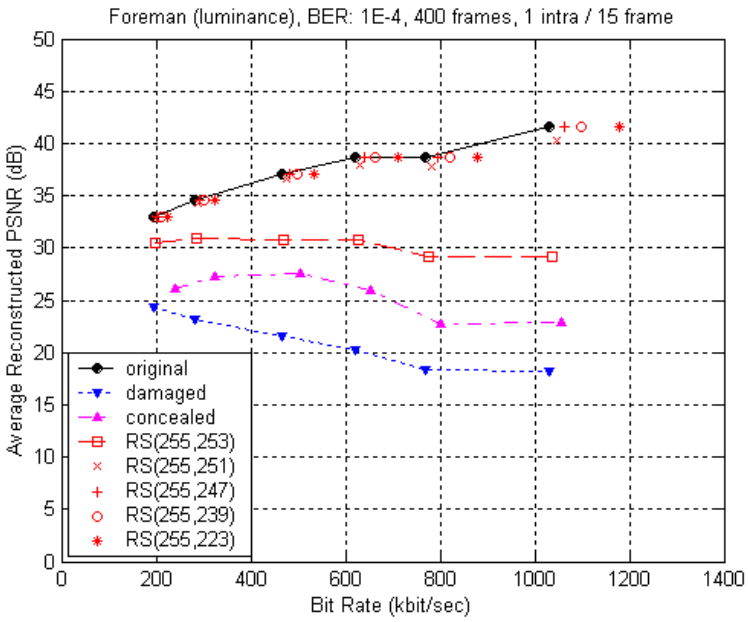


(a)

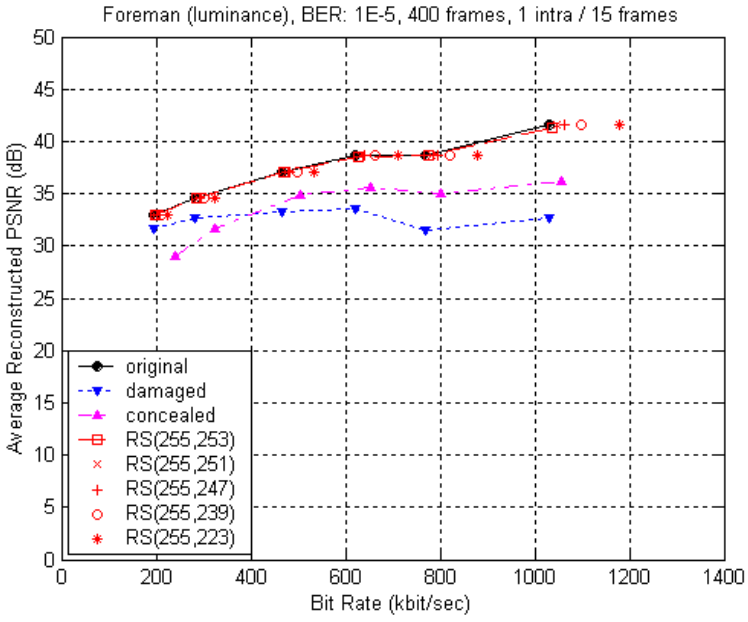


(b)

Figure 5.13. Performance comparison of the proposed system with the Reed-Solomon codes for the *Coast* sequence: average reconstructed PSNR values of all frames vs. channel rate at the BER of (a)10^{-4} and (b)10^{-5}.

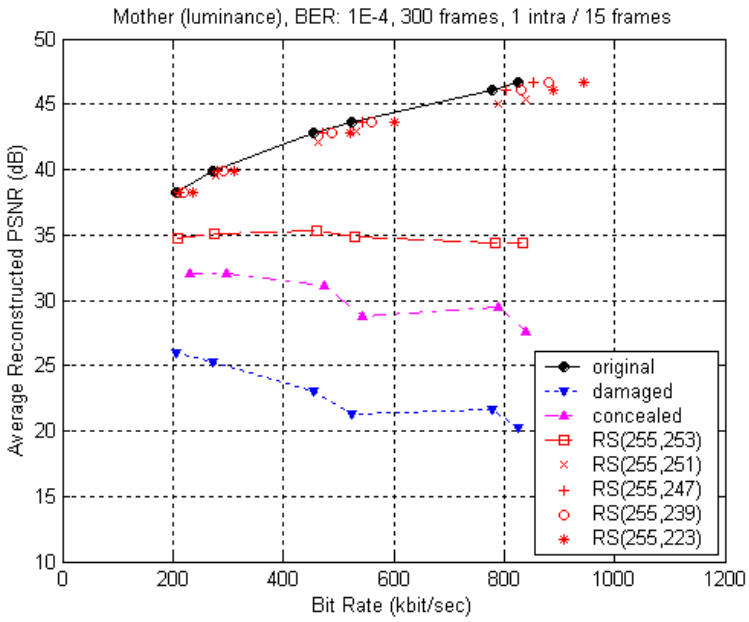


(a)

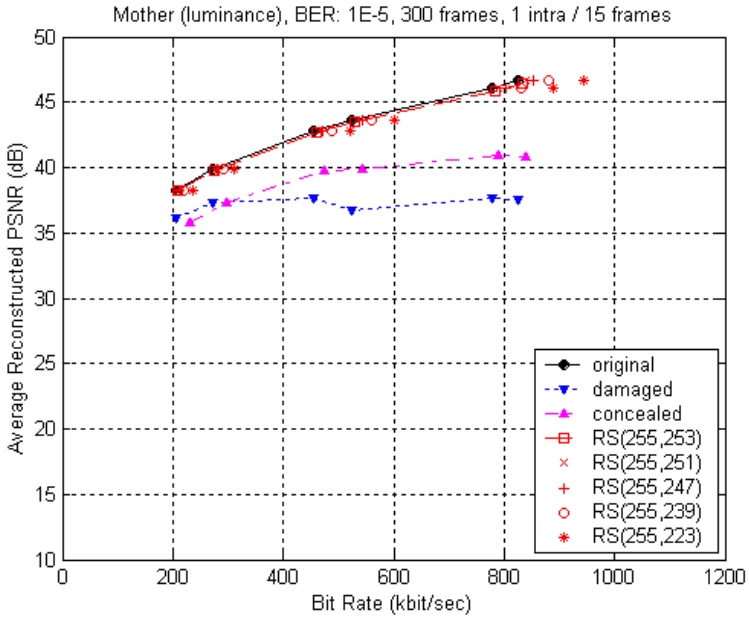


(b)

Figure 5.14. Performance comparison of the proposed system with the Reed-Solomon codes for the *Foreman* sequence: average reconstructed PSNR values of all frames vs. channel rate at the BER of (a)10^{-4} and (b)10^{-5}.

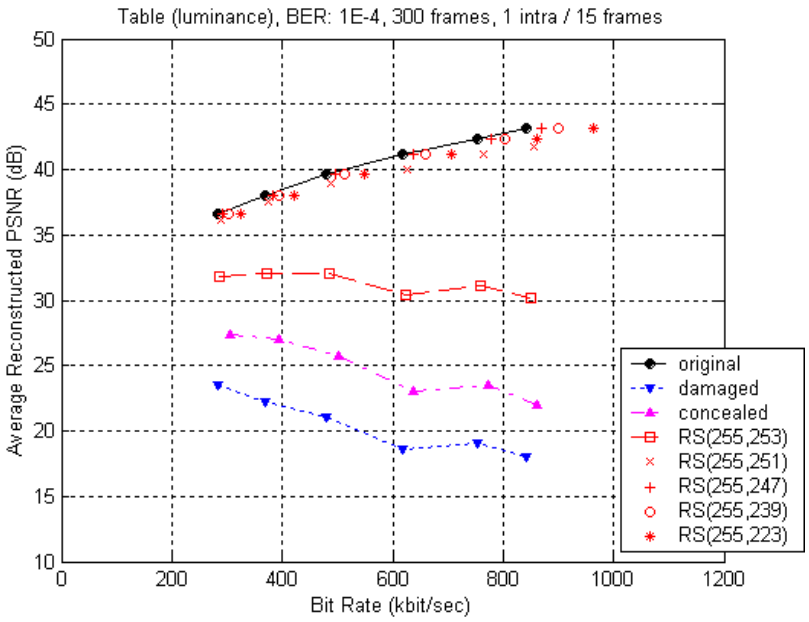


(a)

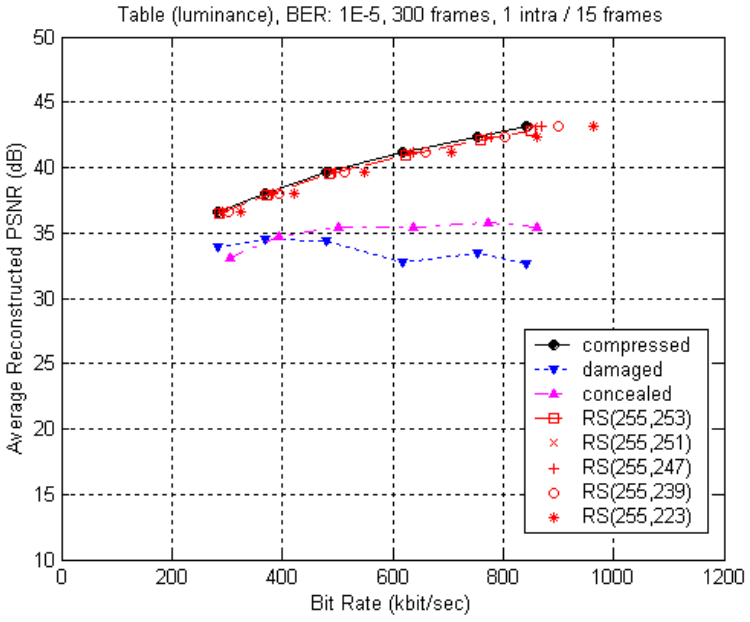


(b)

Figure 5.15. Performance comparison of the proposed system with the Reed-Solomon codes for the *Mother* sequence: average reconstructed PSNR values of all frames vs. channel rate at the BER of (a)10^{-4} and (b)10^{-5}.



(a)



(b)

Figure 5.16. Performance comparison of the proposed system with the Reed-Solomon codes for the *Table* sequence: average reconstructed PSNR values of all frames vs. channel rate at the BER of (a)10^{-4} and (b)10^{-5}.

5.5 Computation Time

The simulations are conducted on a PC with 256 MB RAM, Intel Pentium III 864 MHz CPU, and Windows 2000 operating system. The baseline H.263+ software decodes the bit stream of QCIF *Foreman* sequence encoded at 500 kbit/sec in 23.46 fps. The proposed method decodes the same bit stream in 21.15 fps. Hence, the modifications on the H.263+ codec by the proposed method do not cause significant increase in coding time.

CHAPTER 6

CONCLUSIONS

A novel video error concealment method, which achieves detection, synchronization and reconstruction using data hiding, is proposed. The system combines a number of previous methods in order to obtain better reconstruction quality. In addition to this combination, some novel methods are also proposed to improve the efficiency of the previous methods.

The intensities of the damaged block in intra-frames are recovered by edge-based interpolation from neighborhood blocks as a past-processing method. The edge direction information of the damaged block is transmitted to the decoder by hiding it to a neighbor block's DCT coefficients. It should be noted that all the blocks do not have the same characteristics from reconstruction point of view. Although the edge directional interpolation is superior to the conventional bilinear interpolation, the simulations show that the blocks without a major single edge (such as highly textured areas) cannot be interpolated successfully via edge-based interpolation.

Some errors do not cause large visual degradations on the block and since the interpolation schemes, in these situations, are not able to provide a better reconstruct quality than the current block, a measure of the visual damage of the block is necessary before reconstructing a damaged block. Utilizing a two-bit (overconcealment) parity, obtained from the MSBs of quantized DCT coefficients, is proposed to overcome this problem. Although the performance of overconcealment bits is satisfactory, for the videos containing high frequency components and fast movements, they may not work properly.

Loss of synchronization arises as another problem in intra-frame error concealment. Since the header structure of MB in H.263+ does not provide a

synchronization point to the decoder, once the coefficients are started to be decoded erroneously the decoder can miss the starting point of the next undamaged block. Informing the decoder about the bit length of each block is performed by hiding bit length value of each block into a neighbor block as proposed in a past method. It is observed from the simulations that this method is very effective in error concealment because of successfully preventing error propagation, which causes major visual damages on the image.

In order to conceal the errors, they should be first detected correctly. In addition to the error detection scheme in the H.263+ codec, the hidden data is utilized in the proposed system to determine the errors. The hidden values at the encoder side are checked with the recalculated values at the decoder side. However, this check is incapable of detecting the errors that do not change the edge direction and bit length value of the block. Detecting this type of errors is very important, since they are so likely to destroy the hidden data, although they cause a small visual distortion on the block. Utilizing a 1-bit parity is proposed to overcome this problem, which is neglected in the previous error concealment methods using data hiding.

The single parity bit check detects the small errors, which are missed by the other hidden value comparisons, and verifies the reliability of the hidden data that will be extracted from the related block. While the PSNR loss due to hiding this one bit is negligible, the PSNR, gained by utilizing it, is considerable as observed from the simulations. This observation gives an important clue on the performance of ECC codes on the hidden data, since single-bit parity can be assumed as the simplest ECC.

The errors in inter-frames are row wise concealed by hiding MVs of one row of blocks into the next frame's DCT coefficients of row of blocks as in a previously proposed inter-frame error concealment technique. However, the errors damaging the hidden data rather than the block, like the situation in intra-frames, distort this hidden MV data. If the system does not notice the error in the hidden data, it conceals the damaged blocks in the previous frame by wrong MV data. In order to detect these errors, a checksum is employed for the hidden data in the proposed system. The simulations have shown that utilizing the checksum increased the efficiency of the MV data and the reconstructed PSNR significantly as a result.

The proposed system shows its resilience to errors at higher error-rates, compared to baseline codec. The reason of observing better performance at higher bit-rates is due to finding enough number of non-zero coefficients to hide the required data. If there are not enough coefficients, then the proposed system can not use the hidden data.

Simulations for comparing the proposed system against utilization of ECC after source coding have shown that ECC gives better reconstruction results than proposed system, especially in noisy channels. However, H.263+ bit stream has acquired extra functionalities, such as error detection, resynchronization, and reconstruction, by the proposed system while remaining compatible with the standard decoders. In addition, the usage of some parity bits has increased the efficiency of the hidden data considerably. Thus, if hidden data is protected much more with some kind of ECC, then the proposed system may be improved, which is an ongoing work.

REFERENCES

[1] Yao Wang and Qin-Fan Zhu, “Error control and concealment for video communication: A review,” *Proceedings of the IEEE*, vol. 86, no. 5, pp. 974-997, May 1998.

[2] Min-Cheol Hong, Harald Schwab, Lisimachos P. Kondi, and Aggelos K. Katsaggelos, “Error concealment algorithms for compressed video,” *Signal Processing: Image Communication*, Elsevier, vol. 14, no. 6-8, pp. 473-492, 1999.

[3] Yao Wang, Stephan Wenger, Jiangtao Wen, and Aggelos K. Katsaggelos, “Error resilient video coding techniques,” *IEEE Signal Processing Magazine*, pp. 61-82, July 2000.

[4] Benjamin W. Wah, Xiao Su, and Dong Lin, “A survey of error-concealment schemes for real-time audio and video transmissions over the internet,” *Proceedings of the IEEE International Symposium on Multimedia Software Engineering*, pp. 17-24, December 2000.

[5] Shahram Shirani, Faouzi Kossentini, and Rabab Ward, “Error concealment methods, A comparative study,” *Proceedings of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering*, pp. 835-840, May 9-12 1999.

[6] David Kwon and Peter Driessen, “Error concealment techniques for H.263 video transmission,” *Proceedings of the 1999 IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing (PACRIM ’99)*, pp. 276-279, August 1999.

[7] Stefan Katzenbeisser and Fabien A. P. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House, Inc., USA, 2000.

[8] Ingemar J. Cox, Matthew L. Miller, and Jeffrey A. Bloom, *Digital Watermarking*, Academic Press, USA, 2002.

[9] Mahalingam Ramkumar, Ali N. Akansu, and A. Aydın Alatan, “On the choice of transforms for data hiding in compressed video,” *Proceedings of the 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '99)*, vol. 6, pp. 3049-3052, 15-19 March1999.

[10] Ingemar J. Cox, Joe Kilian, Tom Leighton, and Talal Shamoon, “Secure spread spectrum watermarking for multimedia,” *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1673-1687, December 1997

[11] Joseph J.K. Ó Ruanaidh and Thierry Pun, “Rotation, scale and translation invariant spread spectrum digital image watermarking,” *Signal Processing*, Elsevier, vol. 66, no. 3, pp. 303-317, May 1998.

[12] Min Wu and Bede Liu, “Watermarking for image authentication,” *Proceedings of the 1998 IEEE International Conference on Image Processing (ICIP ’98*), vol. 2, pp. 437-441, 4-7 October 1998.

[13] International Telecommunication Union, Telecommunication Standardization Sector (ITU-T), “Draft text of recommendation H.263 version 2 (“H.263+”) for decision,” 27 January 1998.

[14] Wenjun Zeng and Bede Liu, “Geometric-structure-based error concealment with novel applications in block-based low-bit-rate coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 4, pp. 648-665, June 1999.

[15] David L. Robie and Russell M. Mersereau, “The use of Hough transforms in spatial error concealment,” *Proceedings of the 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '00)*, vol. 4, pp. 2131-2134, 5-9 June 2000.

[16] Peng Yin, Bede Liu, and Hong Heather Yu, “Error concealment using data hiding,” *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01)*, vol. 3, pp. 1453-1456, 7-11 May 2001.

[17] Shao Yafei, Zhang Li, Wu Guowei, and Lin Xinggang, “Reconstruction of missing blocks in image transmission by using self-embedding,” *Proceedings of International Symposium on Intelligent Multimedia, Video, and Speech Processing*, pp. 535-538, 2-4 May 2001.

[18] David L. Robie and Russell M. Mersereau, “Video error correction using steganography” *Proceedings of the 2001 IEEE International Conference on Image Processing*, vol.1, pp.930-933, 7-10 October 2001.

[19] Jae-Won Suh and Yo-Sung Ho, “Motion vector recovery using optical flow,” *Proceedings of the 2000 IEEE International Conference on Consumer Electronics (ICCE ’00*), pp. 234-235, 13-15 June 2000.

[20] Wai-Man Lam, Amy R. Reibman, and Bede Liu, “Recovery of lost or erroneously received motion vectors,” *Proceedings of the 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '93)*, vol. 5, pp. 417-420, 27-30 April 1993.

[21] Luigi Atzori, Francesco G. B. De Natale, and Cristina Perra, “A spatio-temporal concealment technique using boundary matching algorithm and

mesh-based warping (BMA–MBW)," IEEE Transactions on Multimedia, vol. 3, no. 3, pp. 326-338, September 2001.

[22] Yen-Chi Lee, Yucel Altunbasak, and Russell Mersereau, "A temporal error concealment method for MPEG coded video using a multi-frame boundary matching algorithm," *Proceedings of the 2001 IEEE International Conference on Image Processing (ICIP '01)*, vol. 1, pp. 990-993, 7-10 October 2001.

[23] Young H. Jung, Yong-goo Kim, and Yoonsik Choe, "Robust error concealment algorithm using iterative weighted boundary matching criterion," *Proceedings of the 2000 IEEE International Conference on Image Processing (ICIP '00)*, vol. 3, pp. 384-387, 10-13 September 2000.

[24] Mei-Juan Chen and Sen-Yi Lo, "Temporal Error Concealment Using Two-Step Block Matching Principle," *Proceedings of the 2001 IEEE International Conference on Consumer Electronics (ICCE '01*), pp. 172-173, 19-21 June 2001.

[25] Yuan-Chen Liu, Ming-kuan Lee, J. B. Niou, and H. L. Chen, "A novel error concealment technique for MPEG-2 video decoder," International Conference on Consumer Electronics, *Proceedings of the 2001 IEEE International Conference on Consumer Electronics (ICCE '01*), pp. 158-159, 19-21 June 2001.

[26] Mohammed E. Al-Mualla, Nishan Canagarajah, and David R. Bull, "Error concealment using motion field interpolation," *Proceedings of the 1998 IEEE International Conference on Image Processing*, *(ICIP '98)* vol. 3, pp. 512-516, 4-7 October 1998.

[27] M. Al-Mualla, N. Canagarajah, and D. R. Bull, "Temporal error concealment using motion field interpolation," *Electronics Letters*, IEE, vol. 35, no. 3, pp. 215-217, 4th February 1999.

[28] M. E. Al-Mualla, C. N. Canagarajah, and D. R. Bull, "Motion field interpolation for temporal error concealment," *IEE Proceedings - Vision, Image, and Signal Processing*, vol. 147, no. 5, pp. 445-453, October 2000.

[29] Mohammed E. Al-Mualla, C. Nishan Canagarajah, and David R. Bull, "Multiple-reference temporal error concealment," *Proceedings of the 2001 IEEE International Symposium on Circuits and Systems (ISCAS '01)*, vol. 5, pp. 149-152, 6-9 May 2001.

[30] Sang-Hak Lee, Dong-Hwan Choi, and Chan-Sik Hwang, "Error concealment using affine transform for H.263 coded video transmissions," *Electronics Letters*, IEE, vol. 37, no. 4, pp. 218-220, 15th February 2001.

[31] Jie Song and K. J. R. Liu, "A data embedding scheme for H.263 compatible video coding," *Proceedings of the 1999 IEEE International Symposium on Circuits and Systems (ISCAS '99)*, vol. 4, pp. 390-393, 30 May-2 June 1999.

[32] Peng Yin, Min Wu, and Bede Liu, “Robust error resilient approach for MPEG video transmission over internet,” *Visual Communication and Image Processing*, SPIE, vol. 4671, pp. 103-111, January 2002.

[33] Teng Sing Wang, Pao-Chi Chang, Chih-Wei Tang, Hsueh-Ming Hang, and Tihao Chiang, “An error detection scheme using data embedding for H.263 compatible video coding,” Coding of Moving Pictures and Associated Audio, ISO/IEC JTC1/SC29/WG11, MPEG99/N6340, July 2000.

[34] A. Piva, R. Caldelli, V. Cappellini, A. De Rosa, “Data hiding for transmission error detection in H.263 video,” *Tyrrhenian International Workshop on Digital Communications (IWDC 2002)*, Capri, Italy, 8-11 September, 2002.

[35] Alper Koz and A. Aydın Alatan, “Foveated image watermarking,” *Proceedings of the 2002 IEEE International Conference on Image Processing*, *(ICIP ’02)*, vol. 3, pp. 657-660, 24-28 June 2002.

[36] Jae S. Lim, *Two-Dimensional Signal And Image Processing*, Prentice Hall PTR, USA,1990.

[37] Richard E. Blahut, *Theory And Practice Of Error Control Codes*, Addison-Wesley Publishing Company, Inc., 1983.

[38] Brian Chen and Gregory W. Wornell, “Quantization index modulation methods for digital watermarking and information embedding of multimedia,” *Journal of VLSI Signal Processing*, vol. 27, no. 1, pp. 7-33, February 2001.