EIGENSPACE MODEL BASED ERROR CONCEALMENT AND LOW BIT RATE CODING OF FACE SEQUENCES

by

XUEJUN HU

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2005

ACKNOWLEDGEMENTS

The author wishes to express her sincere appreciation to Dr. K. R. Rao, her supervisor, for his continuous guidance and support throughout this research. His understanding and inspiration have made this research productive, challenging and interesting.

The author is grateful to Dr. S. Oraintara, Dr. Q. Liang, Dr. V. Devarajan, and Dr. W.A. Davis who are members of the advisory committee for their generous help and constructive discussions. Special thanks to all the members in Multimedia Processing Research Laboratory, from past to present, for their help at various stages of this research. The author acknowledges financial support from the Department of Electrical Engineering, the University of Texas at Arlington over these years.

Finally, the author would like to express her deepest gratitude to her family for the continuous love, great help and encouragement.

December 16, 2005

ABSTRACT

ERROR CONCEALMENT AND LOW BIT RATE CODING

OF FACE SEQUENCES

Publication No. ______

Xuejun Hu, PhD.

The University of Texas at Arlington, 2005

Supervising Professor: K.R. Rao

The emerging multimedia applications address the increasing demand for novel video coding systems to provide higher compression ratio while maintaining the high quality of the reconstruction. This research makes an effort under such context to investigate the application of principal component analysis (PCA) in the video coding area, especially for error concealment and very low bit rate face coding.

PCA is a well known optimal linear scheme for dimension reduction in data analysis. The central idea of PCA is to reduce the dimensionality of a data set while retaining as much as possible the variation in the data set. Since PCA captures the statistical variations and global information efficiently, it is used in the proposed research to build the model of the target object or range of interest (ROI), and thereby a new model based framework is constructed for very low bit rate face coding and error concealment.

The research focuses on building an efficient and accurate PCA model for very low bit rate coding and effective error concealment. The main limitation of PCA is that it cannot model the data set with large variations efficiently. An adaptive update scheme is investigated in this research to enhance the accuracy and efficiency of the eigenspace model. Computational complexity reduction is another important consideration for real time operation. An incremental mode PCA with missing data for eigenspace updating is proposed. Its effect on the model based error concealment scheme over different quantization levels, loss patterns and loss rates is analyzed. A novel model based and waveform based hybrid coding system aimed at very low bit rate face coding is also presented. Model based coding provides great potential for bit rate savings while model failures and unknown objects can be handled by waveform based coding. The two coding modes are combined under a rate-distortion framework, where Lagrangian cost function is used to determine the most efficient prediction for each block. Simulations show that the system can achieve high compression ratios while maintaining the robustness and generality, which indicate its potential use for videophone application.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

LIST OF ACRONYMS AND NOTATIONS

ADSL	Asymmetric digital subscribe line
ARQ	Automatic retransmission request
AVC	Advanced video coding
BMA	Block matching algorithm
CABAC	Context based adaptive arithmetic coding
CAVLC	Context adaptive variable length coding
CCF	Cross correlation function
CCIR	Consultative committee for international radio
DCT	Discrete cosine transform
DMVE	Decoder motion vector estimation
DPCM	Differential pulse code modulation
EM	Expectation maximization
EREC	Error resilient entropy coding
FDP	Facial definition parameter
FEC	Forward error correction
FFT	Fast Fourier transform
FMO	Flexible macro-block ordering
HDTV	High definition television
HVS	Human visual system
IPCA	Incremental principal component analysis
ISO	International standards organization
ITU	International telecommunication union
KLT	Karhunen-Loeve transform
LLE	Locally linear embedding
MAE	Mean absolute error

MB	Macro block
MDS	Multiple dimensional scaling
MFI	Motion field interpolation
MPC	Mixture of principal components
MSE	Mean squared error
MSR	Maximally smooth recovery
NAL	Network abstract layer
NAL	Network abstraction layer
PCA	Principal component analysis
POCS	Projection onto convex sets
PPCA	Probabilistic principal component analysis
PSA	Principal surface analysis
PSNR	Peak signal to noise ratio
ROI	Region of interest
RTP	Real time transport protocol
RVLC	Reversible variable length coding
SAD	Sum of absolute difference
SKL	Sequential Karhunen-Loeve
SVD	Singular value decomposition
VCL	Video coding layer
VLC	Variable length coding
VQPCA	Vector quantization principal component analysis
$X = UDV^T$	SVD decomposition of matrix X
X	Training matrix X
U	Left eigenspace
D	Diagonal matrix
V	Right eigenspace
P	Matrix composed of additonal columns of vectors
$\bar{x}_i$	Image vector

CHAPTER 1

INTRODUCTION

Video coding is one of the most important topics in multimedia processing and telecommunications [9]. Technology advances and application demands lead to the inevitable merging of telecommunications and computing areas. Under such circumstances, video engineering has become a digital discipline. Representing video material in digital formats requires huge amount of bits, which is very demanding for bandwidth and memory storage, and hence forms a huge challenge for most existing storage and transmission systems. Video compression allows making the most efficient use of available transmission or storage capacity and therefore is the absolute requirement for the growth and success of the low bandwidth transmission and storage of digital video signals.

Besides achieving efficient compression, maintaining as high quality of the reconstructed video as possible is another goal of video coding. In the practical transmission of compressed video, bit errors may occur, which lead to objectionable visual distortion in the decoded video. In order to minimize the visual degradation at the decoder end, many error control techniques [64] have been developed. Some are exercised at the source encoder where redundancy is introduced into the bit stream to make the bit stream more resilient to potential errors, so that an error will not adversely affect the decoder operation leading to unacceptable distortions in the reconstructed video. There are many ways to introduce redundancy into the bit stream. Examples include inserting resynchronization markers and partitioning data into independent segments to isolate the errors, applying modified binary encoding methods such as reversible variable length coding (RVLC) and error resilient entropy coding (EREC) to make the bit stream more robust to transmission errors. Some error control mechanisms are exploited at the transport level. These methods are applied to coded video streams to detect, correct and if necessary, retransmit the damaged data. Examples include applying forward error correction (FEC) for bit errors dominated channel and automatic repeat request (ARQ) for non-real-time data transmission. There are other error control mechanisms [67] which are invoked at the decoder upon detection of errors to conceal the effect of errors.

During the last a few decades, tremendous progress in the visual coding and communication field has been made. However, the emerging new multimedia applications continuously demand higher compression than those provided by the state of the art technologies. For example, wireless cellular video telephony must operate at very low bit rates which can only be achieved through large compression of data. Furthermore strong error control and concealment function must be included to retain the good quality of the reconstruction after the transmission of compressed video content over the error prone wireless channel. To extend multimedia content usage from high bandwidth networks to all types of networks, including those with low bandwidth and high error rates such as internet and wireless channel, there is a need to develop novel compression schemes and novel error control tools so as to provide high compression ratios while maintaining the high quality of the reconstruction.

Principal component analysis (PCA) is a well known optimal linear scheme for dimension reduction in data analysis.[1] The central idea of PCA is to reduce the dimensionality of a data set while retaining as much as possible of the variation in the data set. It has been proven to be an effective approach for pattern recognition, regression and time series prediction.[4][5][6] The focus of this thesis is the application of PCA in the video coding area, especially in low bit rate coding of face sequences and error concealment. The goal of video coding technology is to achieve efficient compression while maintain high quality reconstruction. Besides, real time operation is another requirement for video coding scheme. Hence, the application of PCA in video coding system is investigated to achieve high compression ratios, high quality and expected real time operation.

1.1 Model based error concealment

Various channel and network errors can cause damage or loss of video information during transmission or storage. The consequent distortion can be short time degradation or can completely ruin an image or video signal. Therefore it is necessary to apply error control to minimize the distortion so as to guarantee the reconstruction quality at the decoder end. There are many ways to perform error control. [64] The video encoder can play an important role by embedding a controlled amount of redundancy in the compressed bit stream to make the stream more resilient to errors. Or the channel coder can insert forward error correcting (FEC) or employ automatic retransmission request (ARQ) to apply error control at transport level. This dissertation focuses on error concealment which is another error control mechanism performed at the decoding end.

Error concealment is based on estimation and interpolation procedures that do not require additional information by the encoder. Compared with other error correcting techniques, such as FEC or ARQ, error concealment has the advantage of not requiring extra bandwidth and not introducing additional latency. This property makes it more suitable for real time applications with strict time constraint, such as video telephony and video conferencing.

Error concealment belongs to the general problem of image recovery or restoration. However due to the characteristics of video coding, the resulting error patterns are very particular, and special measures are usually needed to handle such errors. Some prior knowledge about the video content must be used for rebuilding the lost information in all error concealment methods. Conventionally such a prior condition is built in a heuristic manner by assuming smoothness or continuity of the pixel values. In this dissertation, content-based models are investigated. These models are used as prior condition to perform model based error concealment. A content based model can capture the statistical variations of the content more effectively because it is created specifically for video content. A new adaptive model based algorithm is presented for error concealment application. In the proposed algorithm, a modified incremental PCA with missing data is investigated to update the eigenspace model during the video process leading to a more accurate and efficient eigenspace model.

Experimental results demonstrate that the proposed algorithm outperforms conventional intra frame error concealment and this improved performance is stable across different quantization levels, loss patterns and loss rates.

1.2 Model based very low bit rate coding

Future user requirements are anticipated to be dominated by video-driven applications, with demands for a very high degree of flexibility and extensibility.[9] Such demands will require robust and efficient very low bit-rate video coding approaches to support very high quality video. Current video processing technologies and international standards will not be able to cope with such requirements. They either require high bandwidth for high quality transmission, or perform transmission at low bit rates but with low image quality. The development and evolution of alternative video coding techniques and video processing systems are necessary. Model based coding [15] is one promising class of candidate methods. In these methods, a predefined model is known in advance at both encoding and decoding ends. Instead of transmitting the information of pixel values, only a few parameters of the model which are configured to resemble the object effectively are to be encoded and transmitted. Therefore high compression ratio can be achieved. In this thesis, a novel model based and waveform based hybrid coding system aimed at very low bit rate coding of face images in video sequences is presented. The PCA concept is adopted for model based coding, with modifications to cope with video compression. Model failures and unknown objects are handled by waveform based coding employing conventional prediction/transform block-based coding scheme.[9] The two coding modes are combined under rate-distortion framework, where a Lagrangian cost function is used to determine the most efficient prediction for each block. Simulations show that the system can achieve high compression ratios while maintaining the robustness and generality, which indicate its potential use for videophone application.

1.3 Organization

This dissertation will demonstrate how to apply PCA in the video coding area, especially in error concealment and very low bit rate coding. The dissertation is organized as follows. Chapter 2 provides the fundamental knowledge about PCA, video coding and error resilience. Adaptive model based error concealment is discussed in chapter 3. Chapter 4 illustrates the novel hybrid coding system which applies PCA combined with waveform coding to achieve very high compression while maintain robustness and generality. Finally chapter 5 concludes the thesis and gives suggestions for future research.

CHAPTER 2

FUNDAMENTAL KNOWLEDGE

2.1 Principle, computation and applications of PCA

2.1.1 General information

PCA [1] is an important statistical technique to identify patterns in data. Essentially PCA is a linear transformation that determines a new coordinate system for the data set such that the largest variance by any projection of the data set comes to lie along the first axis, the second largest variance lies along the second axis and so on. PCA has the property of being the optimal linear transformation which optimally minimizes reconstruction error under L2 norm.

2.1.2 Description of PCA using the covariance method

PCA is a well known optimal linear scheme for dimension reduction in data analysis. If an image is represented as a long vector, i.e., image is column concatenated into a vector, all images can be viewed as points in the entire high dimensional image space. In this image space, special objects such as face images occupy only a certain small region from other images. (Figure 2.1) Through PCA analysis, a set of directions in the image space along which the variance of the object cluster is largest can be found. These directions define a lower dimensional object space; therefore an object image can be described by a much shorter vector.

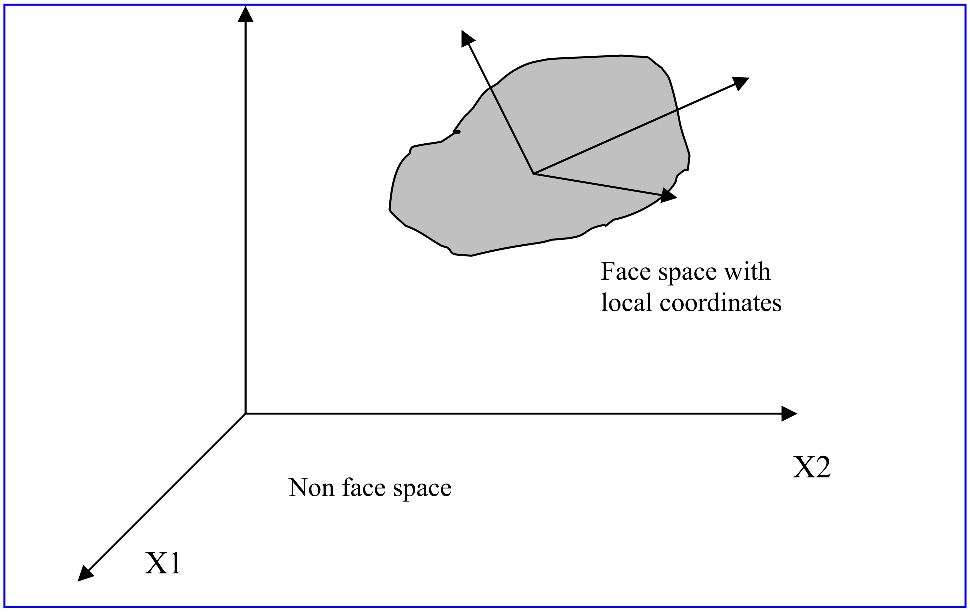


Figure 2.1 The image space and face space coordinate system

The PCA process can be summarized as follows: Let $X_{Tr} = [\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_M]$ be a training data set (of size N^2xM), where $\bar{x}_i \; (i = 1,2,\cdots M)$ are vector representations of images (of size NxN) obtained by concatenating all the columns of the image†.

$$\begin{bmatrix} x_{oo} & x_{o1} & \cdots & x_{0N-1} \\ x_{10} & x_{11} & \cdots & x_{1N-1} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N-10} & x_{N-11} & \cdots & x_{N-1N-1} \end{bmatrix}$$
$$\Rightarrow [x_{00} \quad x_{10} \quad \cdots \quad x_{N-10} \quad x_{01} \quad x_{11} \quad \cdots \quad x_{N-11} \quad \cdots \quad x_{0N-1} \quad x_{1N-1} \quad \cdots \quad x_{N-1N-1}]^T$$

The empirical mean vector $\bar{x}_\mu$ (of size N^2x1) can be found as (2.1):

$$\bar{x}_\mu = \frac{1}{M}\sum_{i=1}^{M}\bar{x}_i \qquad (2.1)$$

†This is lexicographic column ordering of an image of size (NxN)

The covariance matrix C_X (of size N^{2}xN^2) can be expressed as (2.2):

$$C_X = \mathrm{E}\,[\,(\bar{x}_i - \bar{x}_\mu)(\bar{x}_i - \bar{x}_\mu)^T\,] \qquad (2.2)$$

Non-zero vectors $\overline{u_i}$ (of size N^{2}x1) such that

$$C_X\,\bar{u}_i = \lambda_i\,\bar{u}_i \qquad i=1,...,M$$

are the eigenvectors of the covariance matrix while λ_i are the corresponding eigenvalues. These eigenvectors can be considered as a set of features which together characterize the variation among object images such as face images. Each image location contributes more or less to each eigenvector and therefore these eigenvectors are object-like in appearance. In the computer vision problem of human face recognition, these face-like eigenvectors are called as "eigenfaces". Figure 2.2 shows a small sample of training faces [47] and some of the eigenfaces.

Let U= $[\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_k]$ be a matrix (of size N^{2}xk) built with the eigenvectors (each of size N^{2}x1) that correspond to the k largest eigenvalues. The subspace spanned by the eigenvectors of U is called the principal subspace. Using the principal subspace, an image vector $\bar{x}$ (of size N^{2}x1) can be linearly transformed into a k-dimensional vector (usually k is much smaller than the number of pixels of $\bar{x}$, i.e., k<< N^2) by (2.3):

$$\bar{y} = U^T(\bar{x} - \bar{x}_\mu) \quad \text{, where } \bar{y} = [\,y_1, y_2, \ldots, y_k\,]^{\mathrm{T}} \qquad (2.3)$$

Conversely, the original vector $\bar{x}$ can be approximated from its transformed vector $\overline{y}$ as :

$$\tilde{x} = \sum_{i=1}^{k} y_i \bar{u}_i + \bar{x}_\mu \qquad (2.4)$$

(a)

(b)

Figure 2.2 (a) A small sample of training faces (b) Corresponding eigenfaces

2.1.3 Computation of PCA

Eigen analysis of a covariance matrix is the straightforward method to calculate PCA. By introducing the vector $\overline{\phi_i} = \bar{x}_i - \bar{x}_\mu$ (of size N^2x1), and matrix $\Phi = \{\overline{\phi_1}, \overline{\phi_2}, \ldots, \overline{\phi_M}\}$ (of size N^2xM), the covariance matrix C_X can be estimated as (2.5):

$$C_X = \frac{1}{M}\sum_{i=1}^{M}(\bar{x}_i - \bar{x}_\mu)(\bar{x}_i - \bar{x}_\mu)^T = \frac{1}{M}\,\Phi\,\Phi^{\mathrm{T}} \qquad (2.5)$$

C_X is the square symmetric matrix (of size N^2x N^2). The eigenvector of C_X associated with the largest eigenvalue has the same direction as the first principal component. The eigenvector associated with the second largest eigenvalue determines the direction of the second principal component. The sum of the eigenvalues equals to the trace of the square matrix and the maximum number of eigenvectors equals to the number of rows (or columns) of this matrix. To solve for the eigenvalues and eigenvectors of covariance matrix, Householder reduction is first performed leading to a tridiagonal form, followed by the QL algorithm with implicit shifts[2][3].

However there is a computational difficulty with this method in many occasions. The covariance matrix C_X is a two-dimensional N^2 x N^2 array, and determining N^2 eigenvectors and eigenvalues is a very computationally demanding task for typical image size of (NxN) in many applications. Hence a computationally feasible method is needed to determine these eigenvectors. Since only a limited number of principal components are of interest, the relationship between principal component analysis and singular value decomposition (SVD) [2] can be exploited to calculate some of the eigenvectors without the need to compute the covariance matrix.

There is a connection between PCA and SVD. Let X denote an mxn matrix of real value data with rank r, where without loss of generality $m \geq n$ and therefore $r \leq n$. Singular value decomposition of X is defined as :

$$X = UDV^T \tag{2.6}$$

where U is mxn matrix where e columns are orthogonal to one another. V is an orthogonal matrix of size nxn. D is nxn diagonal matrix with elements $\sigma_1,...,\sigma_n$. The existence of the SVD of a matrix can be derived from the eigenvalue decomposition. Consider the matrix $A = X^T X$ (of size nxn). Since A is symmetric, it has real eigenvalues and its eigenvalue decomposition has the form as $A = V\Lambda V^T$, where Λ is diagonal matrix with elements $\lambda_1,...,\lambda_n$. Based on the assumption that A is full rank, a matrix U is constructed as:

$$U = XV\Lambda^{-1/2} \tag{2.7}$$

It is easy to show that orthogonal columns of U are eigenvectors of the matrix XX^T and SVD of X is:

$$X = U\Lambda^{1/2}V^T = UDV^T \tag{2.8}$$

where D is diagonal matrix with elements $\sigma_1,...,\sigma_n$

From the derivation above, it can be seen that there is a direct relation between PCA and SVD in the case where principal components are calculated form the covariance matrix. Each column vector $\overline{u}_i$ (of size $m \times 1$) of matrix U is the eigenvector of product matrix (XX^T), which in fact is the covariance matrix (Here $i = 1,2,\cdots,m$); while each column vector $\overline{v}_k$ of matrix V is the eigenvector of product matrix ($X^T X$), and λ_i, $(i = 1,2,\cdots,n)$, the eigenvalues of $X^T X$, are equivalent to $\sigma_i^{\ 2}$, $(i = 1,2,\cdots,n)$.

Based on this connection, an alternative method to compute PCA exists. Instead of computing the matrix (XX^T) to estimate the covariance matrix, symmetric matrix ($X^T X$) is first calculated and V^T and Λ are obtained by diagonalizing ($X^T X$). Based on the relationship between SVD and PCA, U is derived as :

$$U = XV\Lambda^{-1/2} \tag{2.9}$$

Because the size of matrix ($X^T X$) is much smaller than that of (XX^T), the computational complexity of calculating eigenvectors of ($X^T X$), which leads to the construction of V, is quite acceptable. Therefore a feasible way to calculate the incomplete set of eigenvectors of covariance matrix is done by calculating smaller size matrix V. The fact that the basis vectors are ordered according to descending variance is useful, since only a limited number of eigenvectors are used to perform good quality reconstruction of a large number of different faces.

2.1.4 Applications of PCA

PCA has been widely used in many fields such as face recognition and image compression, and is the common technique for finding patterns in data of high dimension [1].

2.1.4.1 PCA for face recognition [4]

The problem of facial recognition is formulated as the identification of the new face image from the original set. PCA solves this problem by measuring the difference between the new image and the original images, not along the original axes, but along the new axes derived form the PCA analysis. It turns out that these axes work much better for recognizing faces, because the PCA analysis has identified the underlying statistical patterns in the data set. In practice, some of the less significant eigenvectors can be left out and the recognition still performs well.

2.1.4.2 PCA for image compression [5] [6]

PCA for image compression is also known as the Karhunen Loeve transform (KLT). Transform coding is one of the most popular techniques for image compression.

Image data is highly correlated in spatial domain. Transform coding converts the image from the spatial domain into transform domain in order to make them more amenable to compression. Basically, transformed data are nearly or fully decorrelated and energy is packed into a small number of significant coefficients. Therefore if only the elements with high energy are kept, the image can still be restored by an inverse transform with acceptable quality, thus compression is achieved. From practical implementation point of view, the general requirement for the transform basis is that the basis should be signal independent. When compressing a clearly defined class of objects such as images of human faces, the optimal linear transform in a statistical sense such as PCA can still be investigated.

PCA is such a kind of optimal linear transform which aims to find orthonomal basis to compress the images. It involves a mathematical procedure that transforms a number of correlated variables into a smaller number of uncorrelated variables called *principal components*. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible. By ignoring the less important principal components, good approximation can still be obtained by a linear combination of a small number of components, thereby compression is achieved. However, when the original data is reproduced, the images have lost some of the information. This compression technique is said to be lossy.

2.2 Fundamentals of video coding

2.2.1 The importance of video compression

Digital representation of video information has many advantages over traditional analog video. However representing video material in digital form generates huge volume of data which is too large to be handled for most storage and transmission systems. Consider a color video sequence generated using the CCIR 601 [7] format. Each image frame is made up of 720x480 pixels. At the rate of 30 frames per second and 8 bits/pixel per color, this corresponds to a data rate of about 248M bits per second whereas the available ADSL channel bandwidth is only 2Mb/s. Meanwhile we can see that a 4.7 Gbyte DVD can store just 19 seconds of such uncompressed video at this rate. There is obvious capacity shortage of current transmission channel and storage media to meet the bit rate requirement of raw data.

2.2.2 The possibility of video compression

An obvious gap exists between the bit rate demand from raw data and current transmission and storage capacity. The purpose of video compression is to fill this gap. Video compression can be viewed as image compression with a temporal component and is achieved by reducing the redundancies in video sequence. These redundancies can be classified into three types: inter pixel redundancy, coding redundancy and psycho-visual redundancy. [65]

Inter pixel redundancy emerges from the fact that the pixels of an image frame and pixels of successive frames in a video sequence are correlated and therefore value of any given pixel can be predicted from the value of its neighbors. Consequently information carried by individual pixels is relatively small. Much of the visual contribution of a single pixel to an image is redundant. Inter-pixel redundancy can be further divided into two categories: spatial and temporal redundancy.

Spatial redundancy is also called intra frame redundancy. Pixels within one frame are often correlated. Decorrelation of these data can lead to compact representation instead of representing pixels in a frame individually and independently. Predictive or differential coding is based on this observation. By reducing this large amount of redundancy in an image frame a lot of data can be saved in representing the frame.

Temporal Redundancy is also called inter frame redundancy. There are similarities between the successive pictures in the video sequence. As a result the correlation can be removed by predicting a frame from its neighboring frames and coding the differences. Motion compensated predictive coding is the well known temporal prediction scheme for inter frame predictive coding and is used in all the video compression standards.

Coding redundancy is present when the codes assigned to a set of events have not been selected to take full advantage of the probabilities of the events. If an image is coded in a way that uses more code symbols than absolutely necessary, the resulting image is said to contain coding redundancy. It is almost always present when an image is represented with natural binary coding. It is possible to achieve compression by assigning fewer bits to the more probable events than to the less probable ones. This process is commonly referred to variable length coding. Huffman coding and arithmetic coding [65] are the popular examples of variable length coding techniques.

Psycho-visual redundancy originates from the characteristics of the human visual system (HVS) that the HVS does not respond with equal sensitivity to all visual information. Certain information is less important than others based on visual perception. This information is said to be psychovisually redundant and can be eliminated without significantly impairing the quality of image perception. For example, the HVS is much more sensitive to low frequencies than to high ones and so it is possible to compress an image by eliminating certain high frequency components.

2.2.3 Video compression techniques

There are many redundancy reduction techniques which are employed in video codec to remove redundancy and achieve compression [66]. A detailed description is given as follows:

DPCM (Differential Pulse Code Modulation)

This method reduces the redundancy by predicting the value of pixels based on the one or more previously transmitted samples and coding the prediction error. The prediction can be simply formed from the previous pixel or more accurately obtained using a weighted average of neighboring pixels. Due to spatial correlation, the prediction error is typically small and compression can be achieved by assigning shorter code to frequent small prediction errors and longer code to infrequent,larger prediction errors. Following that the quantizer is often included to quantize the prediction error and reduce its precision. Thereby further compression may be achieved

Transform Coding

The aim of transform coding is to reduce the spatial redundancy by mapping the pixels into a transform domain. In transform domain, the image energy is mainly concentrated into a few transform coefficients; hence removing a number of visually insignificant coefficients cannot affect the reconstructed image quality. The transform process itself does not achieve compression. The compression is realized through a lossy quantization process in which the insignificant coefficients are removed, leaving behind a small number of significant coefficients. Transform coding forms the basis of most of the popular image and video compression system.

Motion Compensation and Estimation

Motion compensated prediction is similar to DPCM. The prediction is formed by translating or warping the samples of the previously transmitted reference frame. The resulting motion compensated prediction frame is subtracted from the current frame to produce a residual error frame. Transform coding, quantization and entropy coding usually follow the motion compensated prediction.

Before the motion compensation, motion estimation is carried out to estimate the motion of the moving object. The commonly used motion estimation technique is the block matching algorithm (BMA) [8]. In typical BMA, a frame is divided into predefined blocks of NxN pixels. It assumes that all the pixels within the block have a uniform motion. The process of block matching is to find a candidate block, within a search area in the previous frame, which is the most similar to the current block in the present frame based on certain matching criterion such as cross correlation function (CCF), mean squared error (MSE) and mean absolute error (MAE)

Entropy Coding

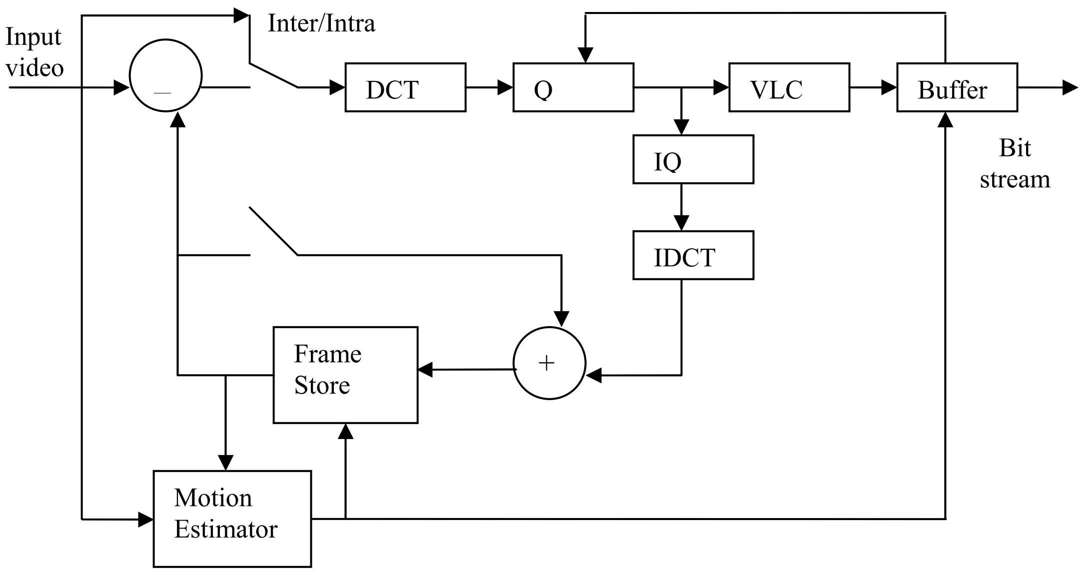
Entropy coding is statistically compression method. There are two types of entropy coding which are employed in the standard video codecs: Huffman coding and arithmetic Coding.

Huffman coding [66] assigns variable number of bits to each symbol based on the statistical distribution of the symbols. Short code words are allocated to common symbols and longer code words are allocated to infrequent symbols. Each code word is chosen to be uniquely decodable. So the decoder can extract the series of variable length codes without ambiguity.

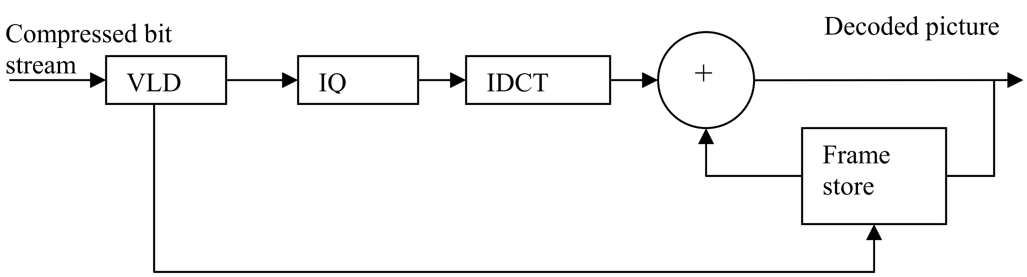
Arithmetic coding [66] maps a series of symbols to a fractional number that is then converted into a binary number and transmitted. Since each symbol is represented by a fractional number of bits and this means that the bits allocated per symbol must be more accurately matched to the statistical distribution of the coed data, arithmetic coding has the potential for higher compression than Huffman coding and plays an important role in the advanced video coding systems.

2.2.4 Generic inter-frame predictive coder [9]

Figure 2.3 shows a generic inter frame encoder which is used in all the standard coding systems.



(a)



(b)

(I)DCT = (inverse) discrete cosine transform   VLC=variable length coder

(I)Q = (inverse) quantization   VLD=variable length decoder

Figure 2.3 Generic inter-frame predictive coding (a) encoder (b) decoder [9]

- Interframe prediction: In this process instead of coding the value of the pixel itself, the difference between pixels in the current frame and their prediction is coded and transmitted. When there is motion, a pixel for prediction in the previous frame has to be displaced by a motion vector. In all the standard codecs, motion compensation is performed for blocks of 16x16 pixels, known as macro block (MB).The motion estimation is carried out only for the luminance part of the picture and a scaled version of the same motion vector is used for compensation of chrominance blocks depending on the picture format.

- Intra/Inter switch: Each MB is either inter frame coded or intra frame coded depending on many conditions such as image type, scene activities and error resilience consideration etc.
- Discrete cosine transform (DCT): all MBs are divided into 8x8 luminance and chrominance pixel blocks, and a 2D DCT is taken.
- Quantizer: The quantizer exploits the human eye characteristics by coarsely quantizing the less sensitive higher frequency DCT coefficients. There are two types of quantizers used in video coding standard. For the AC coefficients and DC coefficient, the quantizer with a dead zone is used. For the DC coefficient of intra MB, the other one without dad zone is applied.
- VLC: The quantized DCT values, the MB motion vector values and the corresponding MB addressed are all variable length coded.
- Inverse Q and IDCT: Through inverse quantization and inverse DCT, the quantized DCT coefficients are converted to difference values and added to their previous picture values to generate a reconstruction of decoded picture. This picture is then used as a prediction for coding the next picture in the sequence.
- Buffer: The variability of picture activity leads to the variable bit rate. Therefore some regulation must be included to achieve constant bit rate for transmitting

the coded video onto fixed rate channels. The buffer is used to accomplish this task by storing and releasing the coded data according to the channel rate.

2.2.5 Current video coding standards

Two standards bodies, the International Standards Organization (ISO) and the International Telecommunications Union (ITU), have developed a series of standards that have shaped the development of the visual communications industry. The aim of the coding standards is to support a particular class of application and to encourage interoperability between equipment and systems from different manufactures.[9]

The ITU has developed H.261 [10], H.262 and H.263 [11] for audiovisual services such as video conferencing. H.261 designed for two-way communication over ISDN lines and supports data rates which are multiples of 64Kbit/s. The algorithm is based on the DCT and uses intraframe and interframe compression. The H.263 was developed for low bit rate communication, with emphasis on bit rates below 64Kb/s. It is based on H.261 with enhancements that improve video quality over modems.

MPEG stands for the Moving Picture Experts Group which is the working group established in 1988 to develop standards for digital audio and video formats. The standards being used or under development include:

MPEG-1

MPEG-1 [12], the first lossy compression scheme developed by the MPEG committee, is still in use today for CD-ROM video compression and is part of early windows media players. The MPEG-1 algorithm uses a combination of techniques to achieve compression, including use of the DCT algorithm to first convert each image into the frequency domain and then process the frequency coefficients to optimally reduce a video stream to the required bandwidth. In addition, level 3 of MPEG-1 is the most popular standard for digital compression of audio-known as MP3, provides about 10:1 compression of audio files at reasonable quality.

MPEG-2

MPEG-2 [13] is the standard designed for the compression and transmission of digital broadcast television. It uses bit rates typically ranging from 1.5 to 15 Mbits/second. MPEG-2 is based on MPEG-1.The most significant enhancement from MPEG-1 is its ability to efficiently compress interlaced video and the new scalability function to offer interoperation among different services. With some enhancements, MPEG-2 is the current standard for high definition television (HDTV) transmission.

MPEG-4

MPEG-4 [14] [15] is the standard for multimedia and web compression. MPEG-4 is based on object-based compression, similar in nature to the virtual reality modeling language. Individual objects within a scene are tracked separately and compressed together to create an MPEG-4 file. This results in efficient compression which is very scalable from very low bit rate to very high. It also allows developers to control objects independently in a scene, and therefore introduce interactivity.

MPEG-7

MPEG-7 [16] [17] is also called the multimedia content description interface. It provides a framework for multimedia content that will include information on content manipulation, filtering and personalization, as well as the integrity and security of the content. Contrary to the previous MPEG standards, which described actual content, MPEG-7 represents information about the content.

MPEG-21

MPEG-21 [21] attempts to provide the multimedia framework which describes the elements needed to build an infrastructure for the delivery and consumption of multimedia content, and how they will relate to each other.

MPEG-4/AVC

This is also called H.264/MPEG-4-AVC [18] [19] [20]. It is a jointly developed standard by video coding experts group (VCEG) and MPEG and has been standardized by the ITU under the H.264 name. H.264 contains a number of features that allow it to provide significantly enhanced compression performance and "network-friendly" packet-based video representation addressing "conversational" (video telephony) and "non-conversational" (storage, broadcast or streaming) applications. It incorporates a video coding layer (VCL), which provides the core high compression of the video content, and a network abstraction layer (NAL), which packages that compressed content for delivery over networks. The VCL design has achieved a significant improvement in rate-distortion efficiency—providing nearly a factor of two in bit-rate savings against existing standards. The NAL designs are being developed to transport the coded video data over existing and future networks such as circuit-switched wired networks, MPEG-2/H.222.0 transport streams, IP networks and 3G wireless systems [64].

2.3 Error resilience

2.3.1 Problem formulation and approaches

A video communication system is typically constructed as shown in Figure 2.4. The video is first compressed by source encoder to reduce the data rate, and then several operations such as packetizing, multiplexing, and channel encoding are performed in the transport coder to convert the compressed bit stream into data packets suitable for transmission. At the receiver end, the inverse process is executed to obtain the reconstructed video signal for display.

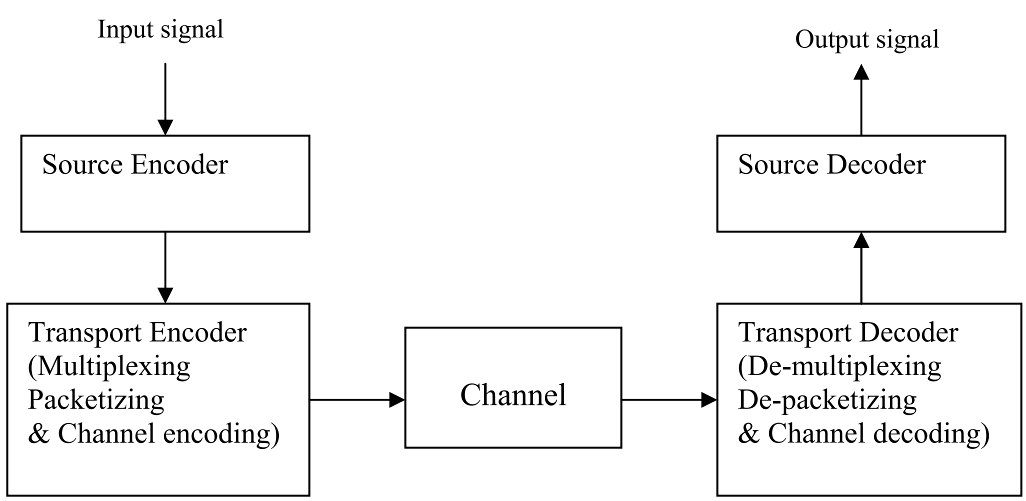


Figure 2.4 Typical video communication system [22]

Data packets may be lost or corrupted during transmission due to imperfection of the channel. The transmission errors can be roughly classified into random bit errors and erasure errors. Different transmission media has different error characteristics which can be described by the combination of network and protocol features [24-30]. The characteristics of current popular networks are summarized as Table 2.1.

Table 2.1 Features of video transmission standards [22]

Application and Standard	Multiplex Protocol	Video Coding	Typical Bit Rate for Video	Packet Size	Error Characteristics
ISDN videophone (H.320)	H.221	H.261 and H.263	64-384kb/s	N/A	Error Free
PSTN Video Phone (H.324)	H.223	H.263	20kb/s	100bytes	Very few bit errors and packet losses
Mobile Videophone (H.324 wireless)	H.223 with mobile extensions	H.263	10-300kb/s	100 bytes	BER==10E-3 to 10E-5, losses of H.223 packets
Videophone over Packet Network (H.323)	H.225/RTP/UDP/IP	H.261, H.263, MPEG-2	10-1000kb/s	<=1500 bytes	BER==0 0-30% packet losses
Cable/Satellite TV	H.222	MPEG-2	6-12Mb/s	N/A	Almost error free
Video conferencing over "Native" ATM (H.310, H.321)	H.222	MPEG-2	1-12Mb/s	53 bytes (ATM cell)	Almost error free

In general, to make the compressed bit stream resilient to transmission errors and to help error prediction and concealment at the decoder, a certain amount of redundancy needs to be added in either the source or transport coder. The classical Shannon information theory [64] states that the source and channel coding can be designed separately, to achieve error free transmission, as long as the source entropy is less than the channel capacity. However, this might be achieved only with a complex code design which is not necessarily implementable in practice. Therefore, a more practical question is formulated as given a fixed capacity channel, and a fixed amount of power, how to allocate them between the source and the channel to get the best result, i.e, the smallest distortion. Various approaches under the name of joint source/channel coding [67] have been developed to solve this problem. All the error resilient encoding methods essentially work under this premise and intentionally make the source coder less efficient than it can be, to prevent transmission errors causing disastrous effect in the reconstructed video quality. For example, in real video codec, the encoder restarts the prediction process periodically. The efficiency of the encoder is thus intentionally sacrificed for limiting the transmission error to small part of a frame, and thereby the missing information can be estimated by spatial and temporal interpolation.

When delivered information is missing due to transmission errors, the decoder can try to estimate it through correctly received samples based on certain inherent correlation. Such techniques are known as error concealment techniques [67]. Error concealment has the advantage of not employing any additional bit rate, but adds computational complexity to the decoder.

Finally the codec and the network transmission protocol must cooperate with each other to take advantage of the added redundancy in the source coder and to facilitate error concealment. For example, there are some bits in the bit streams which are more important than the others, and then a more stringent set of Quality of Service

(QoS) parameters should be assigned to the important part for delivery over a network. To suppress error propagation, the network may also provide a feedback channel to allow the encoder know which part of the reconstructed signal at the decoder is damaged, and not use this part for prediction of future samples.

To summarize, mechanisms devised for combating transmission errors can be categorized into three groups:

1) Those introduced at the source and channel encoder to make the bit stream more resilient to potential errors. Examples include robust entropy coding, multiple description coding, and transport level control etc.[67]
2) Those invoked at the decoder upon detection of errors to conceal the effect of errors. Examples include motion compensated temporal prediction, spatial interpolation, and maximally smooth recovery etc.[31-38]
3) Those which require interactions between the source encoder and decoder so that the encoder can adapt its operations based on the loss conditions detected at the decoder. Examples include reference picture selection based on feedback information, retransmission etc.[67]

2.3.2 Error resilience tools in the state of art coding standard

Error resilience tools widely expand the application range of current video coding standard. In this section, analysis show how the error resilience design in the newest standard H.264/AVC [19] make it possible to achieve acceptable video quality even in highly error prone environments such as wireless communication channel.

The main goal of H.264/AVC standardization effort has been enhanced compression performance and provision of “network-friendly” video representation. To achieve this goal, H.264/AVC has adopted a two-layer structure design, the video coding layer (VCL), which is designed for highly compressed video data, and the network abstraction layer (NAL), which formats the VCL data and adds header information in a manner appropriate for various transportation protocols or storage media. H.264/AVC makes available error resilience mechanisms which are mainly contained in VCL. A brief introduction is presented as follows:

(a) Semantics syntax and error detection

The H.264/AVC video coding standard explicitly defines all the syntax elements, such as motion vectors, block coefficients, picture numbers, and the order they appear in the video bit stream. The main purpose of syntax is to ensure the compliance. Furthermore, it is also an important tool for error detection.

(b) Data partition

Since some syntax elements in the bit stream are more important than others, data partitioning separates the data into different partitions and enables unequal error protection according to their significance in the bit stream. Data partition (DP) A, B and C are defined. Among them, DP A which contains the header information such as MB types, quantization parameters and motion vectors, is the most important one.

(c) Slice structure

The use of slices is another commonly applied method to improve robustness by stopping spatial error-propagation. The MB belonging to a slice can be decoded independently from other slices since no inter-slice dependencies are allowed in H.264. Users can choose different slice structure according to their need. This is called flexible macro-block ordering (FMO) technology. It has been found that the video error concealment schemes perform very well when the lost blocks are arranged in the checker board/scattered blocks fashion or as interleaving of rows.

(d) Parameter set

The parameters, which are used for a group of frames or for a series of slices, usually do not change frequently. Hence various possible combinations of these parameters are classified as parameter sets. The intelligent use of the parameter set greatly enhances error resilience. To signal the parameter configuration to decoder, the encoder just need to transmit the index of the parameter set, instead of the values of parameters themselves. Since only the indices are transmitted, redundant information or extra protection can be added to ensure the reliable transmission of the index. This is the key to using parameter sets to improve error resilience.

(e) Intra block refreshing by R-D Control

Intra block refreshing and I frame insertion are commonly used to stop temporal error propagation when no feedback channel is available. H.264/AVC uses intelligent intra-block refreshing by R-D control. I frame insertion has a generally high bandwidth cost and severe bit rate variations. Consequently it is not advisable to use it for real time and conversational video services. So intra block refreshing is very important for removing artifacts caused by error and inter prediction drift.

(f) SP/SI Synchronization switching frame

SP/SI mechanism is designed for the purpose of video bit stream switching, but it can also be regarded as an important error resiliency feature in network environments with feedback. SP slices make use of motion compensated predictive coding to exploit temporal redundancy in the sequences, like P slices do. Unlike P slices, however, SP slice coding allows identical reconstruction of a slice even when different reference pictures are being used. They aim essentially at bit stream switching, splicing, random access, VCR functionalities and error resilience issues.

(g) Error concealment

The specific schemes suggested for the H.264/AVC standard involve intra and inter picture interpolation. The intra frame concealment scheme uses interpolation based on weighted average of boundary pixels. For inter frame interpolation based concealment, the recovery of lost motion vector is predicted from its neighboring blocks.

(h) Feed back channel

Feed back channel can be efficiently used with long term memory motion compensated prediction for error resilience and can also be used for packet retransmission. The sender can be informed whether the real time transport protocol (RTP) packets have been received or not by the feedback message sent by the receiver. Then the encoder can either select the corrected received frames as the reference frames in the conversational applications or retransmit the lost packets in the stream applications.

CHAPTER 3

EIGENSPACE MODEL BASED ERROR CONCEALMENT

Transmission errors due to channel noise and failures present a big challenge for video communication because these errors can damage the decoded picture leading to unacceptable degradation in video quality.[67] Therefore great emphasis is put on error control and error concealment research which is of special importance in the increasing cases of video transmission over error prone channels such as mobile and internet channels.

The proposed research assumes that the error region is detected and located in advance. Hence the focus is on the error concealment by decoder post-processing, the replenishment of the lost video content at the decoder end. In comparison with other error correcting techniques, such as forward error correction (FEC) or automatic retransmission request (ARQ), error concealment has the advantage of not requiring extra bandwidth and not introducing additional latency, which make it more suitable for real time applications, such as video telephony and video conferencing.

All the error concealment schemes reconstruct the lost video data based on a certain prior knowledge. The proposed error concealment method employs PCA to model the statistical structure of video content in the range of interest (ROI) and uses this model as prior knowledge to replenish the lost data. Using a trained model as prior knowledge reflects the statistical variation and captures the global information more effectively. Therefore better error concealment effect can be expected compared with conventional error concealment methods which are performed based on general prior knowledge. The accuracy and efficiency of the statistical model are the key parts of this method. Fixed PCA cannot model the data set with large variations efficiently; hence a novel adaptive PCA scheme is proposed to enhance the accuracy of the eigenspace model on line. The proposed method updates the eigenspace using previous frame if it is received correctly and current damaged frame where the damaged parts of the frame are treated as missing data. The updating is carried out in incremental mode which is suitable for real time applications due to its computational efficiency and low requirement for storage memory. This research proposes the incremental updating with missing data method and its application in the novel adaptive PCA scheme to build an accurate and efficient eigenspace model for error concealment.

This chapter is organized as follows. Section 3.1 introduces existing error concealment technology. Section 3.2 shows the fixed eigenspace approach for error concealment and analyzes its limitations. An adaptive eigenspace scheme for error concealment is presented in section 3.3. Section 3.4 describes the incremental updating with and without the missing data. Simulation results are shown in section 3.5. Conclusions and discussion of further research are presented in section 3.6.

3.1 Existing error concealment techniques

We have introduced error control and error resilience in Chapter 2 and described the overall picture of combating the transmission error and improving the image quality. The proposed research focuses on the error concealment. All the error concealment schemes reconstruct the lost video data based on a certain prior knowledge. Most existing error concealment techniques build such a prior condition by assuming smoothness or continuity of the video data in different domains thereby the exiting error concealment methods can be mainly sorted into two categories: spatial error concealment and temporal concealment.

3.1.1 Spatial error concealment

Spatial error concealment methods [64] recover the lost image content by assuming that images are smooth spatially in nature. Based on this smoothness assumption, different constraints and criteria are designed leading to different algorithms such as spatial domain interpolation, interpolation with edge detection, projection onto convex sets and maximally smooth recovery etc. All the methods under this category only make use of the spatial smoothness property and are mainly targeted for still images or for intra coded blocks in video.

Spatial Domain Interpolation [31] [32]

Spatial domain interpolation is a kind of intuitive approach. It interpolates the lost block from surrounding blocks by assuming the correlation among these blocks. In [31], Hemami and Meng proposed an algorithm which performs error concealment on each 8x8 block. The lost coefficients/pixels are estimated by linear combination of the same coefficients/pixels in adjacent blocks. Assuming that the reconstructed block connects smoothly to its neighbors, weights used for interpolation are set to minimize the squared difference between pixels across block boundaries. If surrounding blocks are damaged too, recovered surrounding blocks are used and iterative recovery is executed until there are no further changes to any damaged blocks. Aign and Fazel [32] proposed another method which estimates the lost pixel value within one macroblock from its four 1-pixel-wide boundaries. Two schemes of this method are proposed for interpolation. In the first block based scheme, a pixel is interpolated with the two pixels from its two nearest boundaries, as shown in Figure 3.1(a). The interpolation formula is described as:

$$
\begin{aligned}
b_1(i,k) &= \frac{d_T b_{L2}(i,N) + d_L b_{T3}(N,k)}{d_L + d_T} \\
b_2(i,k) &= \frac{d_T b_{R1}(i,1) + d_R b_{T4}(N,k)}{d_R + d_T} \\
b_3(i,k) &= \frac{d_B b_{L4}(i,N) + d_L b_{B1}(1,k)}{d_L + d_B} \\
b_4(i,k) &= \frac{d_B b_{R3}(i,1) + d_R b_{B2}(1,k)}{d_R + d_B}
\end{aligned}
\tag{3.1}
$$

where i, k=1, 2, …, N

The second macroblock based scheme, shown in Figure 3.1(b), interpolates each pixel of the lost macroblock with the adjacent pixels in all four boundaries as :

$$
\begin{aligned}
mb(i,k) = \frac{1}{d_L + d_R + d_T + d_B}[&d_R mb_L(i,2N) + d_L mb_R(i,1) + \\
&d_B mb_T(2N,k) + d_B mb_T(1,k)]
\end{aligned}
\tag{3.2}
$$

where i, k=1, 2, …, 2N

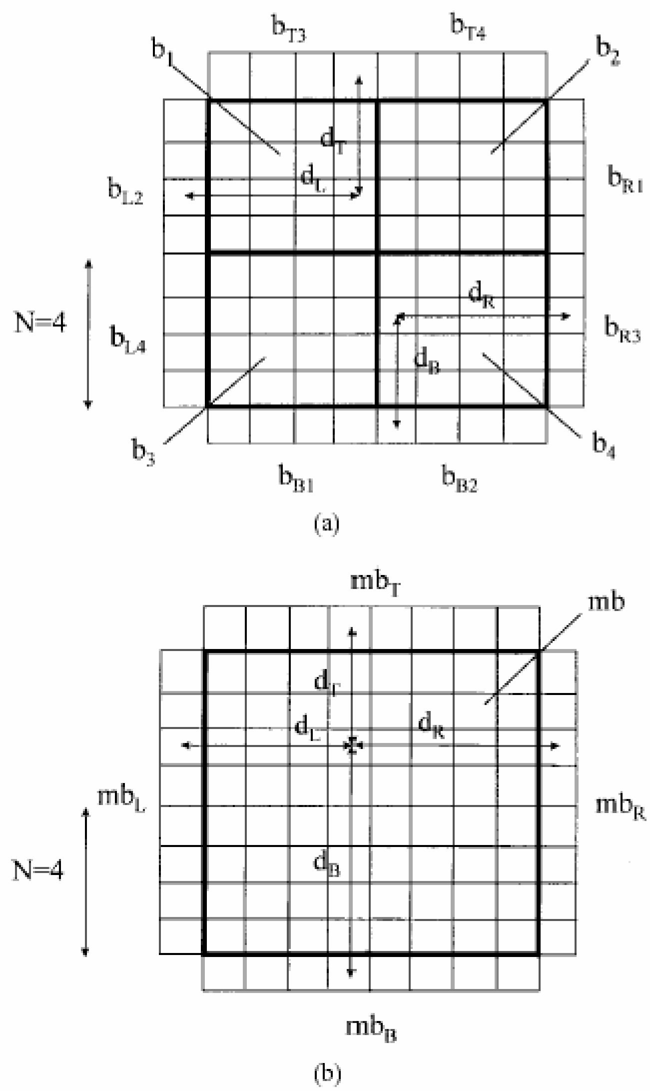


Figure 3.1 Pixel domain interpolation (a) block-based scheme (b) macroblock-based scheme [32]

Interpolation with Edge Detection [33]

Interpolation by surrounding pixels will blur edges running through the damaged blocks. Since edge integrity plays an important role in human visual perception, algorithms are developed to improve the performance of interpolation in edge areas by detecting the edges first and then applying directional interpolation or filtering along the edge directions. Edge directions are estimated based on gradient, and classified into eight directions. For each of these classified directions, a series of one-dimensional interpolations are carried out along that direction. Multiple edges may run through the damaged block. Hence, multiple interpolation blocks are reconstructed using different edge directions. These multiple interpolations are subsequently combined together in such a way that all the strong features of each reconstruction are extracted to get the final error concealed block (Figure 3.2).

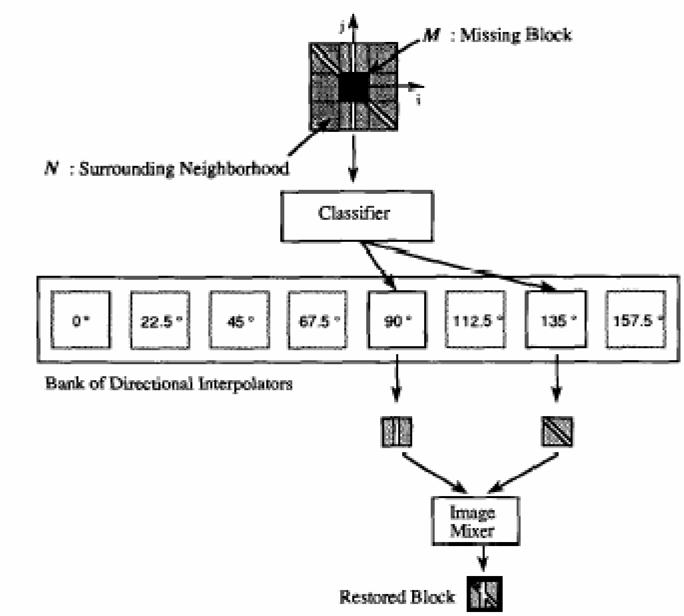


Figure 3.2 Illustration of multiple directional interpolation algorithm [33]

Maximally Smooth Recovery (MSR) [34]

MSR uses the correctly received DCT coefficients and surrounding blocks to reconstruct the damaged block based on the smoothness property of images through energy minimization. The formulated problem is to find the lost coefficients through received ones. Since it is an ill-posed problem, additional constraints have to be imposed to confine the possible solution set. In general, image blocks are smoothly connected with one another and with samples in the surrounding blocks. Therefore MSE selects the solution that maximizes the smoothness criteria. Two kinds of smoothness measures are used. Type 1 smoothing constraint is imposed between every two adjacent samples across the boundary, as shown in Figure 3.3(a). Type 2 smoothing constraint is imposed on each pixel in the direction toward its nearest boundary which is shown in Figure 3.3(b). The resulting image is the maximally smooth image among all those reconstructions with the same coefficients and boundary conditions.

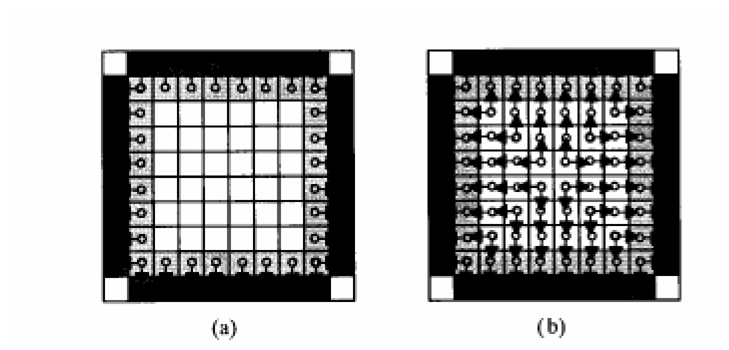


Figure 3.3 Illustration of two smoothing constraints [34]

Projection onto Convex Sets (POCS) [35]

Error concealment methods under POCS [41] framework iteratively use the smoothness constraints and other constraints of pixel values and frequency values to arrive at an optimal rebuilding of lost data. The priori properties of typical video images that are often used are:

1) Smoothness - requires reconstructed samples to be smoothly connected with adjacent image samples
2) Edge continuity - requires that edges be continuous
3) Consistency with known values - requires that correctly receive pixel values not be altered by restoration process, and that restored values lie in a known range (e.g. [0-255])

POCS is realized as shown in Figure 3.4 [35]. Missing blocks, together with 8 surrounding blocks are transformed into frequency domain using FFT [1]. The block is classified as either a monotone block or an edge block. This classification procedure is similar to multi-directional interpolation approach. Based on the block classification results, adaptive filtering is applied: low-pass filter for smooth regions and band-pass filter for regions with edge. Adaptive filtering is the first projection operator which realizes smoothness and edge continuity constrains. After inverse FFT, pixels in the reconstructed missing block are truncated to the integer range 0-255. For the blocks that are correctly received, the pixel values are maintained. This procedure is the second projection operation which realizes constraints of consistency with known values. These two projection operations are applied iteratively until convergence is arrived.

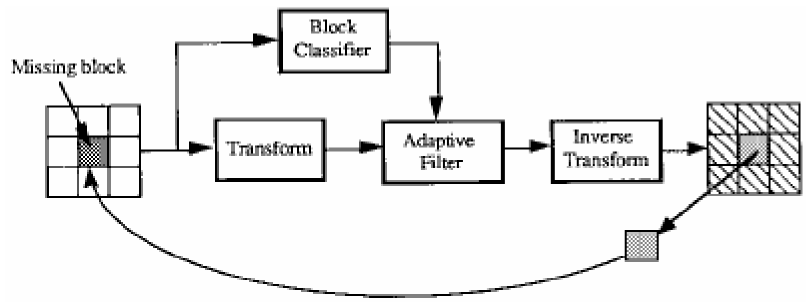


Figure 3.4 Adaptive POCS iterative restoration process [35]

3.1.2 Temporal error concealment [67]

The temporal error concealment methods exploit the temporal continuity property and conceal the damaged block in current frame using blocks from the previous frame. The simplest approach is to replace the damaged block with the spatially corresponding block in the previous frame. For the video sequence with large motions, adverse visual artifacts will be produced. Significant improvements can be achieved by replacing the damaged block with the motion compensated block in the previous frame. A problem with temporal interpolation is that it requires knowledge of the motion information, which can be lost as well. Therefore techniques to estimate the lost coding mode and motion vectors are widely investigated.

The boundary matching algorithm (BMA) is the basic algorithm [36] to estimate the lost motion vector. A set of candidate motion vectors are first built up by the following method:

1) Set the motion vectors to zeros, this works well for video sequences with relatively small motion.
2) Use motion vectors of the corresponding block in the previous frame
3) Use the average of the motion vectors from spatially adjacent blocks
4) Use the median of motion vectors from the spatially adjacent blocks

The estimated motion vector is then chosen from the set of candidate vectors depending on which one produces the smallest boundary variation.

Many other algorithms are extensions to the BMA. Decoder motion vector estimation (DMVE) [37] treats the loss of motion vectors as a motion estimation problem. Motion field interpolation (MFI) [38] estimates the motion vectors from neighbors with single or multiple reference frames. Furthermore, Lee et al. [68] extended translational block motion to an affine transform for motion compensated error concealment.

3.2 Fixed eigenspace error concealment

The proposed error concealment is model based error concealment which builds a priori by training a context-based model for object or region of interest (ROI) and uses this model to recover any missing information. The model for error concealment is constructed on line by using training images from the video sequence which is being decoded, or can be constructed off line by using a training set from database. Since the model is built specifically for certain objects, the statistical variation can be captured more efficiently. Therefore better performance can be expected compared with conventional smoothness and continuity constrained error concealment techniques.

The accurate model is the first requirement for model based error concealment. The question is how to build the model efficiently? We turn to dimensionality reduction technology and select principal component analysis (PCA) as the model building method. PCA identifies the principal directions along which the original data set has the largest variations. These identified directions are eigenvectors of covariance matrix of data. The original data vector can be approximated without large reconstruction error by a linear combination of a few "best" eigenvectors, those corresponding to the largest eigenvalues. The eigenspace composed of these selected eigenvectors is the model which is used for our error concealment scheme.

The fixed eigenspace based error concealment is first examined. For fixed eigenspace based error concealment, once the model is built up, it will not be changed and will be used throughout the error concealment process. The algorithm of the fixed eigenspace based error concealment can be discribed by Figure 3.5.

This scheme is based on the following assumptions:

1) The ROI has been detected and extracted by some segmentation techniques [40]. In object oriented coding scheme such as that in the MPEG-4 [14] standard, ROI information is contained in the coded stream. Therefore it is easy to build the model and implement error concealment. Under such circumstances, model based error concealment is especially useful.
2) A set of training images from an image database or previously decoded frames which are received correctly is previously known and PCA is performed on the training set to build the eigenspace models off line or on line

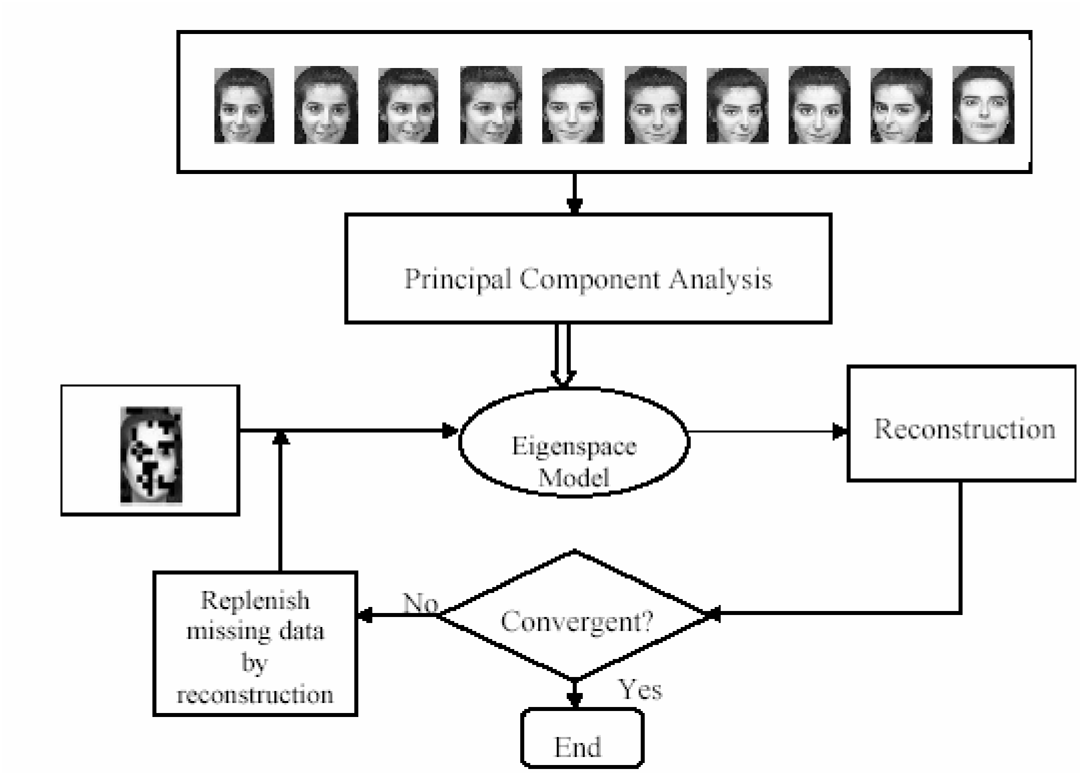


Figure 3.5 Fixed eigenspace model based error concealment

Once the eigenspace is built up, POCS framework is adopted to find the best error concealed image. The entire error concealing procedure includes two steps:

Step 1: The damaged image is projected into eigenspace to obtain the reconstruction. The reconstruction is a linear combination of eigenvectors; therefore set of reconstructions can be viewed as convex set 1 (C_1).

Step 2: The missing data is replenished using reconstruction. For the pixels that are received correctly, their values are maintained. The constraint of consistency with known values is known to be a convex set [41], denoted as convex set 2 (C_2). C_2 contains all signal vectors $\vec{x}$ in n-dimensional real space R^n with some pixels equal to known values. This can be expressed as :

$$C_2 = \{\vec{x} \in R^n : x_i = k_i, i \in I\} \tag{3.3}$$

where x_i is the ith component of vector $\vec{x}$, and k_i are known constraints in a given index set I. The projection P_2 onto the convex set C_2 is given by :

$$[P_2\vec{x}]_i = \begin{cases} k, & i \in I \\ x_{i,} & otherwise \end{cases} \tag{3.4}$$

The iteration of step1 and step 2 projects the damaged data into the convex set 1 and convex set 2, leading to the convergence to the overlapped parts of C_1 and C_2, which has optimal error concealment effect. (Figure 3.6).

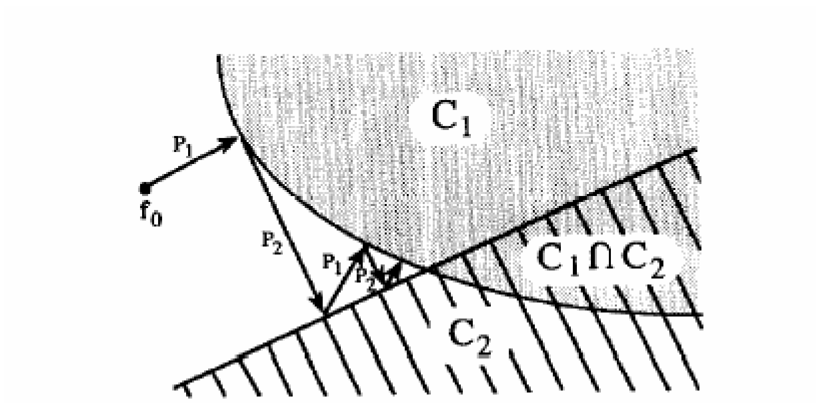


Figure 3.6 Illustration of projection onto convex sets [41]

In order to apply model based error concealment, the employed model should be able to capture the statistical variations in the object appearance accurately and efficiently. Some experiments have been carried out to evaluate the performance of fixed eigenspace based error concealment. Three images with randomly damaged blocks are used for testing. Four iterations are executed under POCS framework for each image. The simulation results are shown in Figures 3.7-3.9.

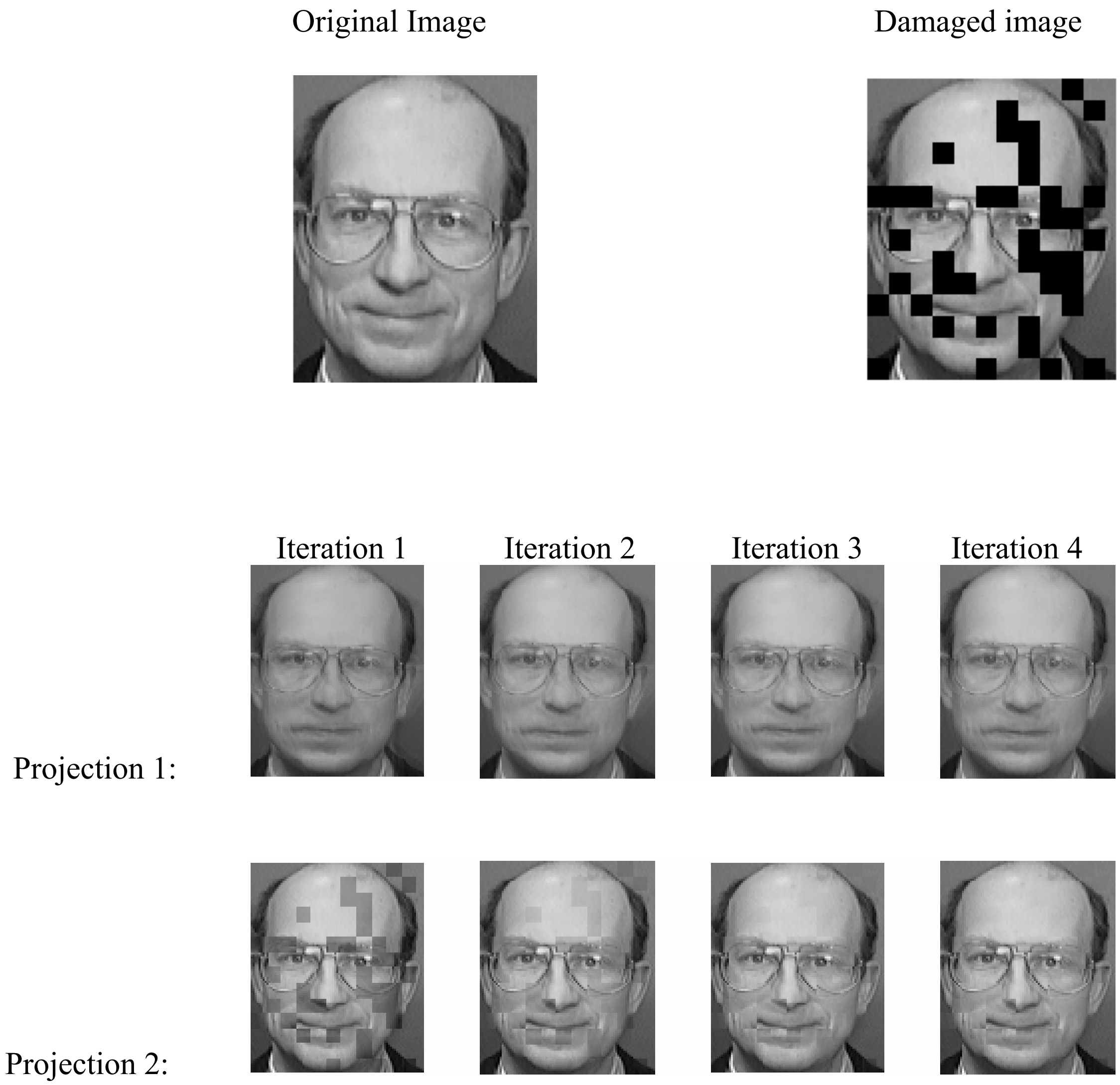


Figure 3.7 Fixed eigenspace error concealment with fair recovery

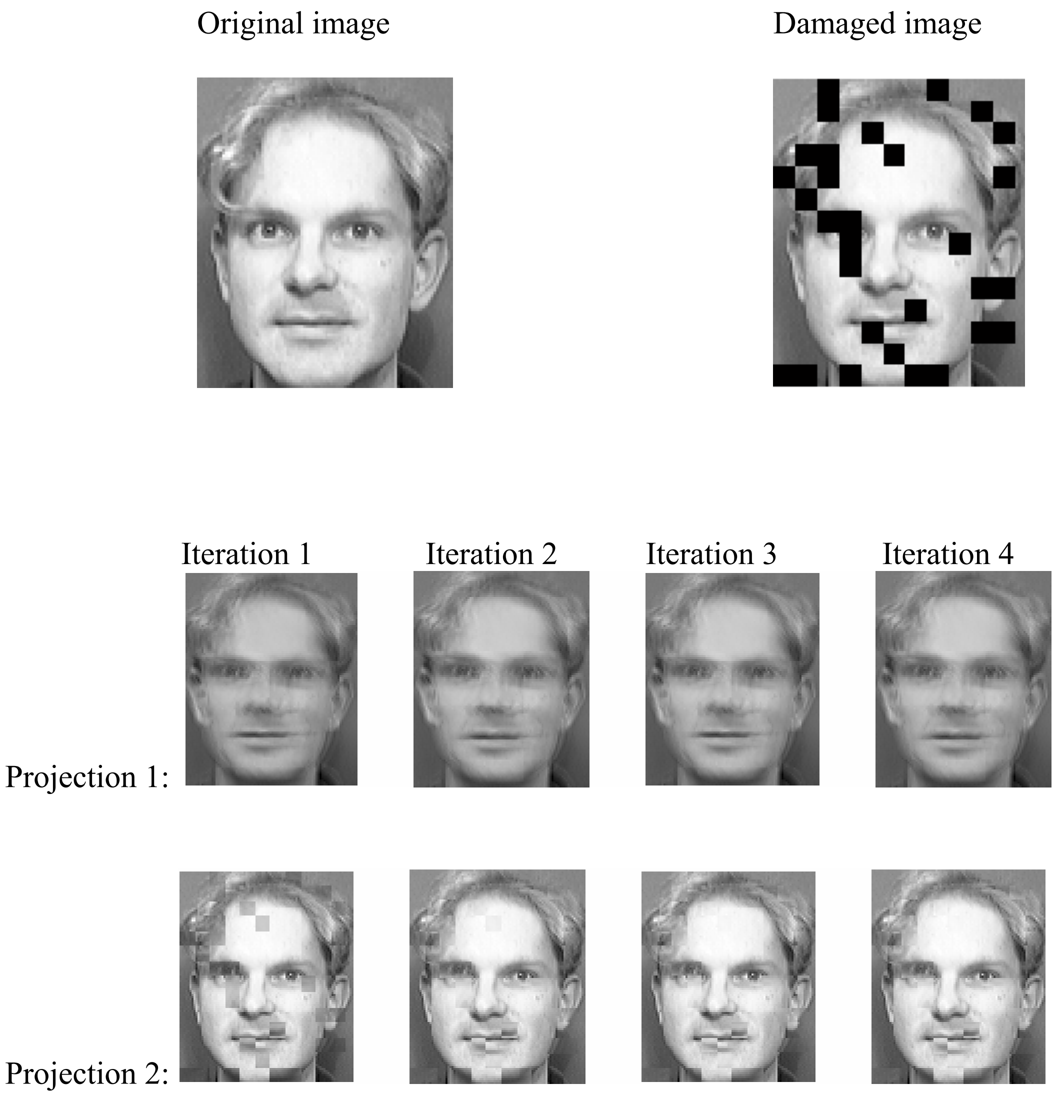


Figure 3.8 Fixed eigenspace error concealment with bad recovery

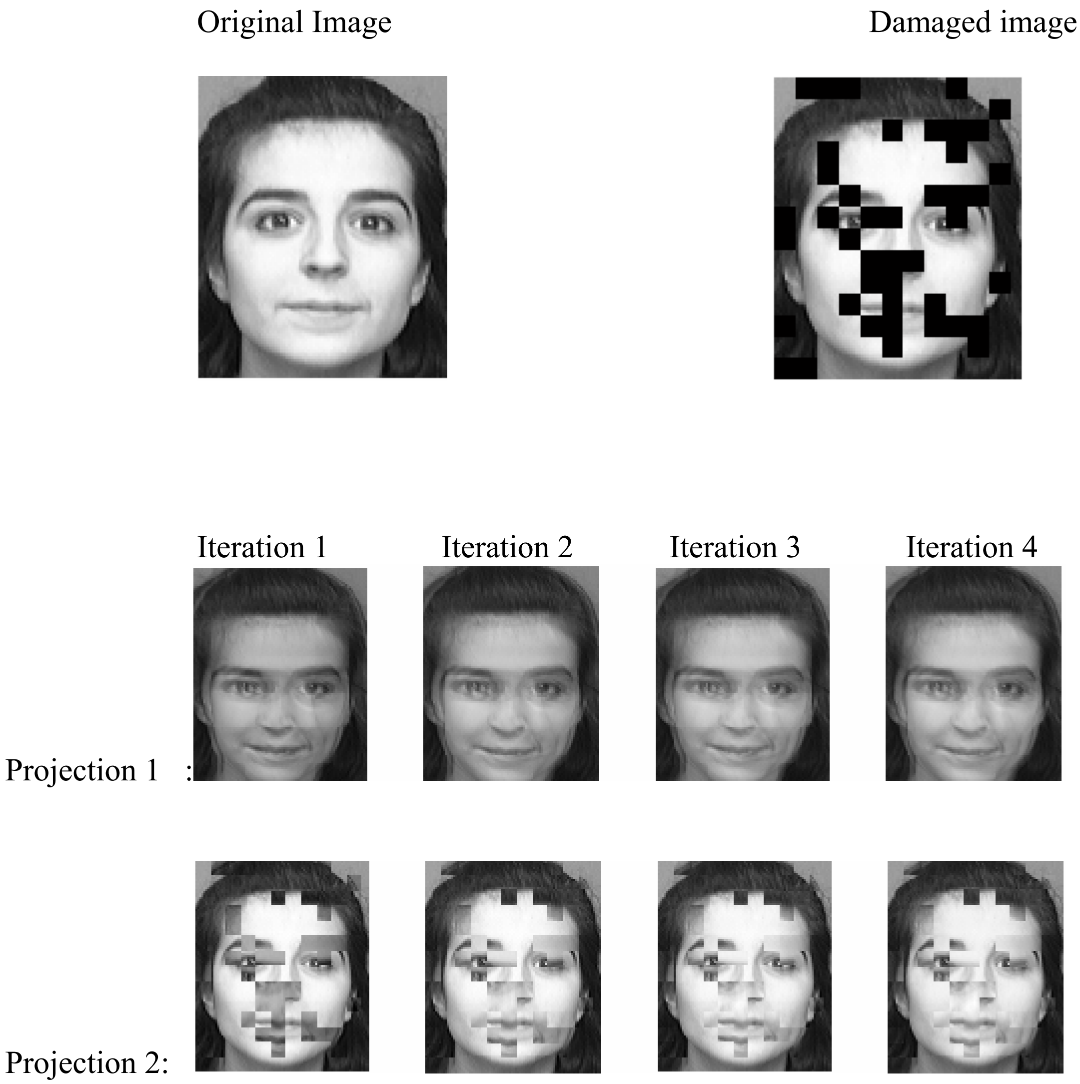


Figure 3.9 Fixed eigenspace error concealment with very bad recovery

Through subjective evaluation it can be seen that good error concealment cannot be guaranteed for each frame and overall performance of fixed eigenspace error concealment is not satisfying. The underlying reason is that the fixed eigenspace cannot capture the large variations in the video sequence.

3.3 Adaptive eigenspace error concealment

The efficiency and accuracy of the model are critical to ensure the satisfactory performance of model based error concealment. Fixed eigenspace is inefficient to model the data set with large variations such as the image frames in a video sequence and therefore its application is limited. In order to make a model based error concealment applicable, accuracy and efficiency of the eigenspace model have to be improved.

Many linear and nonlinear extensions to PCA for model improvement are examined. Nonlinear extensions include principal surface analysis (PSA) [42], multi-dimensional scaling (MDS) [43] and locally linear embedding (LLE) [44]. PSA tries to model the data clusters using parameterized surfaces instead of the hyper-planes that PCA uses. MDS attempts to preserve local relationships by conserving pair wise distance between data points during the dimensionality reduction. LLE adopts the similar approach as MDS. The linear extensions to the PCA are also examined. Among these extensions are vector quantization PCA (VQPCA) [45], probabilistic PCA (PPCA) [46] and mixture of principal components (MPC) [39]. VQPCA first partitions the data set into clusters based on smallest reconstruction error criterion. The parameters of each cluster are then updated using local PCAs, and this process is iterated until convergence of the parameters. Unlike VQPCA which actually employs hard partitioning for clustering, PPCA applies soft partitioning for clustering while training the local PCAs at the same time. Then a mixture of such PPCAs is used to represent the data. MPC automatically models the data using a mixture of eigenspaces. Instead of optimizing the likelihood of observing the data given in the model, which is the method in PPCA, the MPC parameters are chosen to minimize the overall reconstruction error. All these extensions can improve the accuracy of the eigenspace model, but the improvements are still limited and not stable for all kinds of video sequences. Furthermore, the computational intensity is too high.

Fixed eigenspace model is examined again. A drawback of this method is that it did not utilize the history information. Hence a novel error concealment scheme which adopts incremental PCA approach to update the eigenspace on line is proposed. The updated eigenspace is adapted to the frame variations in video sequence thereby a more accurate model for error concealment is obtained. In this section focus is placed on adaptive updating scheme. The question of how to design the computationally efficient updating algorithm will be discussed in the next section.

The immediate previous frame and the current frame which is to be repaired contain the most relevant information for updating. Temporal correlation exits in a video sequence and the immediate previous frame is usually highly correlated with the current frame. If it is used for updating, the updated eigenspace could catch the variations of next frame in advance. The current frame, although partially corrupted, has the correctly received part which contains the most relevant information for error concealment. Therefore it is always used to update the eigenspace thereby constructing an eigenspace which relates most closely to the current frame that is to be repaired.The flowchart proposed adaptive eigenspace based error concealment is shown in Figure 3.10.

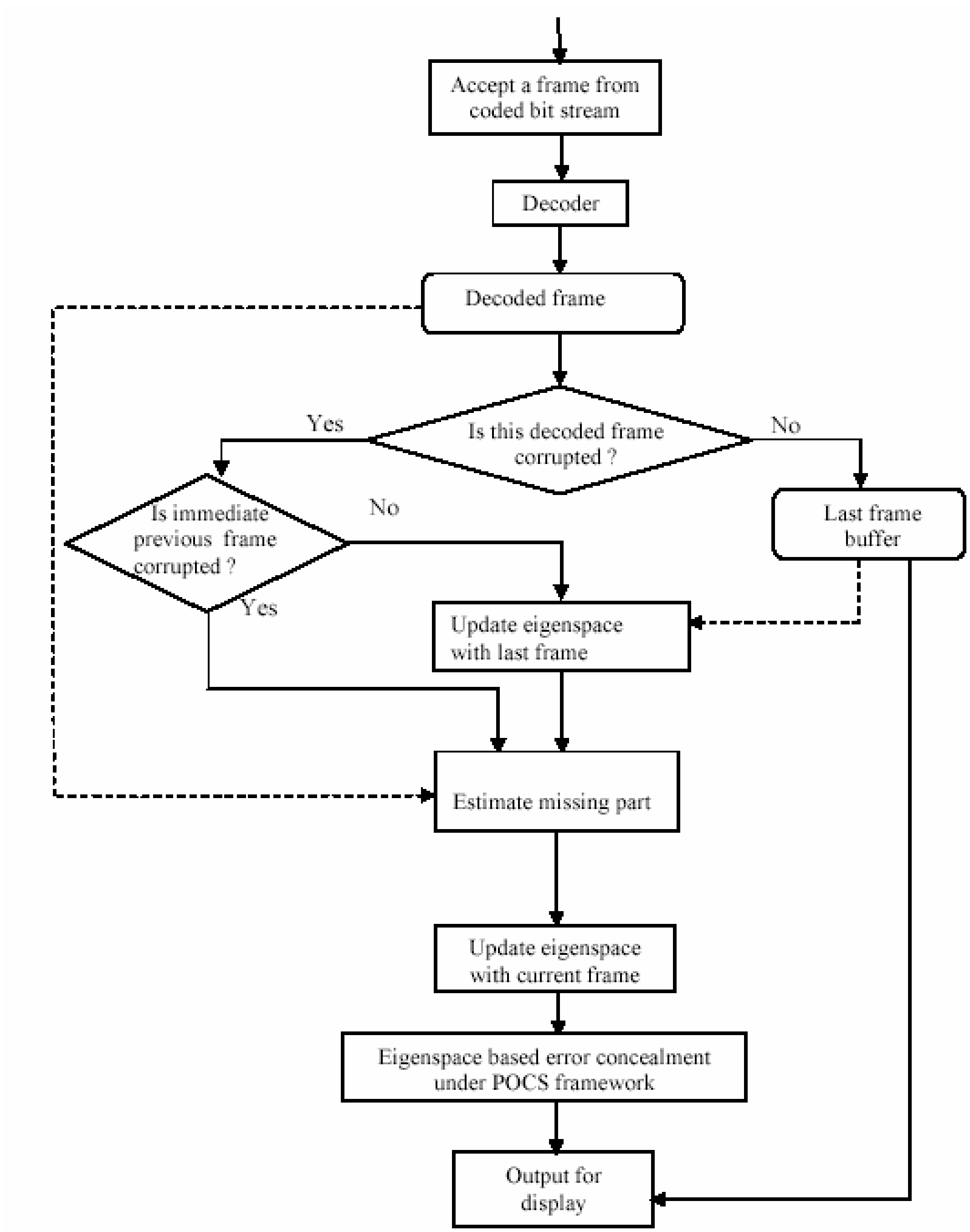


Figure 3.10 Adaptive eigenspace model based error concealment

The decoder receives the bitstream from the channel and reconstructs each frame. The decoded frame then goes through error detection. If there is no damaged part in this decoded frame, it will be saved in “last frame buffer” and it could be used to update the eigenspace model for replenishing next damaged frame. After error detection, the “clean” frame is sent to the display. If the decoded frame is spotted as damaged frame, it will be repaired. First, the distance between the frame in the “last frame buffer” and the current frame is checked. If the distance is equal to one, then the immediate previous frame is clean and eigenspace is updated using this immediate previous frame. Otherwise, this updating will be skipped and then we proceed to next updating stage which updates the eigenspace with the current corrupted frame.

In order to reduce the opposite effect of corrupted part as small as possible, the damaged parts in current frame are estimated roughly before current frame is used for updating. The estimation will be explained in next section.

After estimation, the eigenspace is updated with refurbished current frame. Once the eigenspace is updated, eigenspace based error concealment under POCS frame work is performed. This procedure is the same as the one described in the last section. Stages 1 and 2 are performed iteratively until convergence is arrived and the damaged part is concealed.

Computational complexity is an important consideration for real time operation at the decoder end where the memory resource and computational power are both restricted. Computational economy is the necessary requirement for the eigenspace updating algorithm. The incremental PCA with missing data is the efficient updating solution adopted in the proposed scheme which will be discussed in section 3.4.

3.4 Incremental PCA for eigenspace updating

Eigenspace is conventionally computed by batch mode PCA. Given a set of training vectors, PCA determines the eigenspace spanned by the principal components which are calculated as the eigenvectors of the covariance matrix. Linear combination of these eigenvectors provides the optimal approximation of the original data vectors in the least squares sense. Discarding some eigenvectors corresponding to small eigenvalues will not introduce large distortion in reconstruction; therefore a small set of eigenvectors with large eigenvalues can model the data space reasonably well.

In batch mode PCA, all the training vectors are used simultaneously to compute the eigenspace model. The requirement for computation power and storage memory is demanding. Specially in order to update a subspace of eigenvectors with another image, all the images have to be kept in memory, and re-compute the entire decomposition from scratch every time. The implementation cost and operation delay rule it out from real-time application. Incremental PCA does not need all training vectors at once and can update the existing eigenspace model sequentially by adding new vectors. Thus incremental PCA is selected as the technique for eigenspace updating.

3.4.1 Existing incremental PCA methods

Three methods from the literature [48-51] to incrementally update the dominant singular subspaces of a matrix A are to be discussed. These methods are characterized by the different updating techniques and the different forms of the results at the end of each step. One kind of these methods is characterized by the production of a factorization in SVD-like form, consisting of two orthonormal bases and one nonnegative diagonal matrix. The other kind of method produce orthnomal bases for the singular subspaces along with a small, square matrix. This matrix contains the current singular values, along with rotations that transform the two bases to the singular bases.

"Broken arrowhead" matrix based method

In [48], Gu and Eisenstat propose a stable and fast algorithm to update the SVD when appending a single column or row to a matrix with a known SVD. The kernel step in this algorithm is the efficient tridiagonalization of a rank i+1 "broken arrowhead" matrix, having the form as :

$$B = \begin{bmatrix} \Sigma_i & z \\ 0 & \rho \end{bmatrix} = \begin{bmatrix} \sigma_1 & & & \xi_1 \\ & \ddots & & \vdots \\ & & \sigma_i & \xi_i \\ & & & \rho \end{bmatrix} \tag{3.5}$$

In this algorithm the SVD of the structured matrix is related to a function of a special form, that allows efficient evaluation using the *fast multipole method,* thereby the SVD of B is computed stably and efficiently in $O(i^2)$ computations instead of the $O(i^3)$ computations required for a dense SVD. Although the object of their algorithm is to find an updated complete SVD, and does not concern the tracking of the dominant space, their work is still considered as the foundation for other algorithms that track only the dominant space.

Chandrasekaran et al. [49] propose an algorithm for tracking the dominant singular subspace and singular values, called the eigenspace update algorithm (EUA). The EUA was the first algorithm to adaptively track the dominant subspace, the algorithm can be stated formally as follows:

$U = A_1 / \|A_1\|, V = 1, \Sigma = \|A_1\|$

For i = 2 to N

$[U\Sigma V^T A_i] = \hat{U}\hat{\Sigma}\hat{V}^T$

Find k such that $\hat{\sigma}_k > \sigma \geq \hat{\sigma}_{k+1}$

Let U equal the first k columns of $\hat{U}$

Let V equal the first k columns of $\hat{V}$

Let Σ equal the leading kxk principal sub matrix of $\hat{\Sigma}$

End

All vectors corresponding to the singular values lower than some user-specified threshold are truncated. The SVD of $[U\Sigma V^T A_i]$ can be obtained either via a standard dense SVD algorithm or by utilizing the arrowhead matrix based method. Note that the arrowhead-based method is only possible if a single row or column is used to update the SVD. Levy and Lindenbaum's algorithm [50] allows more efficient inclusion of multiple rows or columns.

Sequential Karhunen-Loeve algorithm

Levy and Lindenbaum [50] proposed an algorithm for incrementally computing the basis for the dominant left singular subspace. The algorithm, named sequential Karhunen-Loeve (SKL), essentially executes a sequence of SVD updating steps, leading to a low dimensional KL-basis of an image sequence. For each step a block of columns is allowed to be brought in and the block size is optimized to minimize the overall complexity of the algorithm, assuming the number of columns per block is controllable. While this work concerns finding the KL basis (the dominant left singular basis), the technique can be modified to compute a low-rank factorization of the matrix without dramatically affecting the performance.

The core of the SKL algorithm is based on partitioning the SVD of a large matrix into two steps. It starts by calculating the SVD of the first block of image data. Then at every step, another block of columns is added and SVD is calculated using "portioned R-SVD" algorithm. Consider the following identity:

$$\hat{B} = [B \mid P] = \underbrace{[U \mid \widetilde{P}]}_{\hat{U}} \underbrace{\begin{bmatrix} D & U^T P \\ 0 & \widetilde{P}^T P \end{bmatrix}}_{\hat{D}} \underbrace{\begin{bmatrix} V^T & 0 \\ 0 & I \end{bmatrix}}_{\hat{V}} \tag{3.6}$$

The incoming block of vectors P (of size *m*x*l*) is separated into components $U^T P$ and $\widetilde{P}^T P$, which are projections of P onto the current dominant subspace U and the subspace $\widetilde{P}$ which is orthogonal to the current dominant space. Next calculate the SVD of $\hat{D}$ as

$$\hat{D} = \widetilde{U}\widetilde{D}\widetilde{V}^T$$

where $\widetilde{U}$ is the matrix composed of the orthogonal left singular vectors, $\widetilde{D}$ is diagonal matrix whose diagonal elements are the singular values of $\hat{D}$ and $\widetilde{V}$ is an orthonormal matrix whose columns are the right singular vectors. Clearly the SVD of $\hat{B}$ is

$$\hat{B} = \hat{U}(\widetilde{U}\widetilde{D}\widetilde{V}^T)\hat{V}^T = (\hat{U}\widetilde{U})\widetilde{D}(\widetilde{V}^T\hat{V}^T)$$

Finally, the rank of the dominant space is determined, based on a user specified threshold and the noise space is truncated.

For each step, the SVD of $\hat{D}$ is computed in negligible complexity $O(k^3)$, where k is the maximal number of columns for the U matrix, but the formation of the dominant part of $\hat{U}\tilde{U}$ requires computation of $2m(k+l)k$. Combined with the formation of $\hat{U}$ from U and P in $4m(k+l)k$, this yields a total complexity of $2mn(k^2+3lk+2l^2)/l$ to process the entire matrix. It is also shown that, assuming a fixed size for k, a block size l that yields a minimal operation count can be determined. The running time is optimized by a block size $l = k/\sqrt{2}$. The authors made qualitative claims about the convergence of the approximation under certain assumptions, but they give neither quantitative explanation nor rigorous analysis.

Incremental PCA for classification (IPCA for classification)

Hall, Marshall and Martin [51] devised a method for incrementally computing eigenspace model in the context of using them for classification. Assume an eigenspace model is constructed over N observations $\bar{x}^i \in R^N$. By calculating the mean vector $\bar{x}_u$, a set of eigenvectors, associated eigenvalues from the covariance matrix, and the number of observation N. Typically only p of the eigenvectors and eigenvalues are kept, therefore the constructed eigenspace model is denoted as $\Omega = (\bar{x}_u, U_{np}, \Lambda_{pp}, N)$,where $\bar{x}_u$ is the mean vector, U_{np} is the dominant eigenspace with p left eigenvectors, Λ_{pp} is diagonal matrix with p diagonal singular values and N is the number of observations.

When the (N+1)th new sample is added, the mean vector $\bar{x}_u$ is easily updated to the new one as (3.7):

$$\hat{\bar{x}}_u = \frac{1}{N+1}(N\bar{x}_u + \bar{x}_{N+1}) \tag{3.7}$$

The new left eigenspace $\hat{U}$ is obtained by adding a new vector $\hat{\bar{h}}$ into the exiting left eigenspace U and applying a rotational transform R as :

$$\hat{U} = [U \quad \hat{\bar{h}}]R \tag{3.8}$$

$\hat{\bar{h}}$ is defined as (3.9):

$$\hat{\bar{h}} = \begin{cases} \dfrac{\bar{h}}{\|h\|^2} & if \ \|h\|^2 \neq 0 \\ 0 & otherwise \end{cases} \tag{3.9}$$

where

$$\bar{h} = (\bar{x}_{N+1} - \bar{x}_u) - U\bar{g} \tag{3.10}$$

$$\bar{g} = U^T(\bar{x}_{N+1} - \bar{x}_u) \tag{3.11}$$

The eigenproblem, after adding $\bar{x}_{N+1}$ is:

$$\hat{C}\hat{U} = \hat{U}\hat{\Lambda} \tag{3.12}$$

where $\hat{C}$ is the new covariance matrix which can be proven to be:

$$\hat{C} = \frac{N}{N+1}C + \frac{N}{(N+1)^2}\bar{y}(\bar{y})^T \tag{3.13}$$

Here $\bar{y}$ is set to $\bar{x}_{n+1} - \bar{x}_u$.

Substitution of $\hat{C}$ into eigenproblem equation gives (3.14).

$$(\frac{N}{N+1}C+\frac{N^2}{(N+1)^2}\overline{y}(\overline{y})^T)[U \quad \overline{\hat{h}}]R=[U \quad \overline{\hat{h}}]R \quad \hat{\Lambda} \tag{3.14}$$

Left multiply of (3.23) by $[U \quad \overline{\hat{h}}]^T$ leads to :

$$[U \quad \hat{h}](\frac{N}{N+1}C+\frac{N^2}{(N+1)^2}\hat{y}(\hat{y})^T)[U \quad \hat{h}]R=R \quad \hat{\Lambda} \tag{3.15}$$

From (3.15), R is derived as the solution to the eigenproblem of the following form:

$$DR=R\hat{\Lambda} \tag{3.16}$$

where

$$D=\frac{N}{N+1}\begin{bmatrix}\Lambda & \overline{0}\\ \overline{0}^T & 0\end{bmatrix}+\frac{N}{(N+1)^2}\begin{bmatrix}\overline{g}\,\overline{g}^T & \gamma\,\overline{g}\\ \gamma\,\overline{g}^T & \gamma^2\end{bmatrix} \tag{3.17}$$

and

$$\gamma=\overline{\hat{h}}^T\overline{y} \tag{3.18}$$

This algorithm explicitly accounts for a change in origin as well as a change in the number of eigenvectors in the basis set. This character makes it especially useful for classification applications

None of algorithms described above deal with the missing data. Two commonly used methods dealing with the missing data problem are to replace the missing elements with the mean or an extreme value. However such approaches are no longer valid when a significant portion of data vector is missing. The SVDimpute algorithm [52] employs expectation-maximization (EM) method to obtain the estimations of the missing values as follows:

E-step: Applying singular value decomposition to obtain the eigenvectors. Since the SVD can only be performed on complete matrices, all missing values in matrix is substituted by row average in the initial stage.

M-step: A missing value j in vector i is estimated by first regressing the vector against the k most significant eigenvectors and then use the coefficients of the regression to reconstruct j from a linear combination of the k eigenvectors. The jth value of the vector i and the jth values of the k eigenvectors are not used in determining these regression coefficients.

The two steps are iterated until the total change in the matrix falls below the empirically determined threshold of 0.01.The iterative nature of this algorithm makes it very time consuming, furthermore it works effectively only for a matrix with a small parts of missing data.

Wiberg [53] proposed a Gauss-Newton method to solve the missing data problem. It is a batch mode algorithm which requires solving large pseudo-inverse matrices.

3.4.2 Proposed algorithm for eigensapce updating with missing data

An updating scheme with incomplete data for model based error concealment is proposed. It can be shown that this scheme is computationally stable and efficient. The algorithm shares the same updating principle with the algorithm in [50], but it is modified to compute the left eigenspace which is spanned with the principal components. The missing data is estimated with the fewest standard deviations from the origin with respect to the data observed so far. Suppose an eigenspace is already built from a set of training images. The existing eigenspace will be updated with newly incoming vector which is the current damaged frame to be repaired, where the damaged part will be estimated before updating. The entire procedure formulates two problems: updating scheme and missing data estimation.

The underlying principle of the adopted updating scheme can be explained as follows. Let X_i denote the matrix $[\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_i]$ (of size $N^2 \times i$), where $\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_i$ are vector representations of training images (each of size $N^2 \times 1$) obtained by concatenating all the columns of the image and ε be a given tolerance. The X_i can be reconstructed to ε accuracy by $U_i D_i V_i^T$. That is $\left\| X_i - U_i D_i V_i^T \right\| \leq \varepsilon$, where $\| \ \|$ denotes standard Euclidean 2 –norm. For the matrix X_i,

$$\|X_i\| = \max_{\|x\|=1} \|X_i x\|$$

U_i and V_i are matrices whose columns are the left and right eigenvectors of covariance matrix respectively. D_i is the diagonal matrix whose diagonal elements are the eigenvalues of the covariance matrix. The initial $\{U_i, \quad D_i, \quad V_i\}$ can be computed by computing the eigen composition of $X_i^T X_i$ (of size $i \times i$). When the new image vector $\bar{x}_{i+1}$ is acquired, the eigenspace is updated and the updating is based on the following identity:

$$X_{i+1} = [X_i \mid \bar{x}_{i+1}] = \underbrace{[U_i \mid \bar{u}]}_{\hat{U}} \underbrace{\begin{bmatrix} D_i & U_i^T \bar{x}_{i+1} \\ \bar{0}^T & \bar{u}^T \bar{x}_{i+1} \end{bmatrix}}_{\hat{D}} \underbrace{\begin{bmatrix} V_i^T & \bar{0} \\ \bar{0}^T & 1 \end{bmatrix}}_{\hat{V}}$$

$\hat{U} = [U_i \mid \bar{u}]$ is columns orthonomal matrix.

The incoming vector $\bar{x}_{i+1}$ is decomposed into components $U_i^T \bar{x}_{i+1}$ and $u_i^T \bar{x}_{i+1}$. $U_i^T \bar{x}_{i+1}$ is the projection of $\bar{x}_{i+1}$ onto the orthogonal basis U_i and $u_i^T \bar{x}_{i+1}$ is the projection of $\bar{x}_{i+1}$ onto the subspace u_i which is orthogonal to U_i. The decomposition is shown in Figure 3.11.

$\hat{D}$ is a *broken arrowheaed* matrix, whose SVD can be computed quickly using the techniques suggested by Gu and Eisenstat [48]. After calculating the SVD of $\hat{D} = \tilde{U}\tilde{D}\tilde{V}^T$, the SVD of X_{i+1} can be represented as $X_{i+1} = \hat{U}(\tilde{U}\tilde{D}\tilde{V}^T)\hat{V}^T = (\hat{U}\tilde{U})\tilde{D}(\tilde{V}^T\hat{V}^T)$. Then the updated left eigenspace is obtained by $\hat{U}\tilde{U}$. This procedure is equal to the rotation of the subspace, which is illustrated in Figure 3.12. The rank of the dominant eigenspace is determined by user specified threshold.

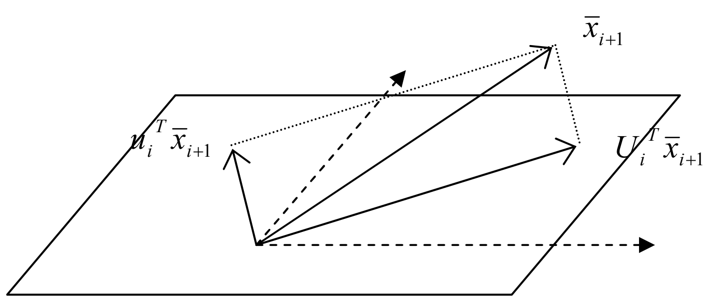


Figure 3.11 A vector is decomposed into components within and orthogonal to the existing dominant eigenspace

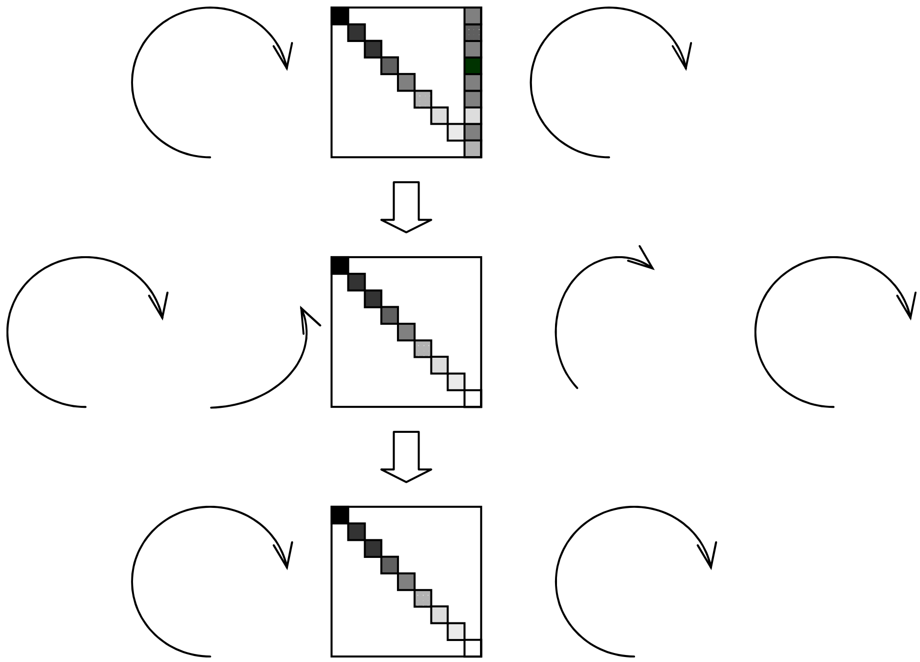


Figure 3.12 Visualization of updating procedure. The *broken arrowheaded* matrix $\hat{D}$ is diagonalized and the subspaces are counter rotated to preserve equality

Based on the above principle, the computation of the proposed algorithm is summarized as follows:

Input: current eigenspace U_i, current diagonal matrix D_i, new input image vector $\overline{a}_{i+1}$

Output: new eigenspace U_{i+1}, new diagonal matrix D_{i+1}

Algorithm:

This algorithm increases the dimension of the subspace by one each time. After the updating, we can discard the least significant principal vector to preserve the dimension of the subspace. The initial value of the eigenvectors and the coefficients can be obtained by applying batch PCA on a small set of images. The current eigenspace

U_i, current diagonal matrix D_i can be initial values calculated by batch mode PCA, or the updated values based on the previous eigenspace U_{i-1} and previous diagonal matrix D_{i-1}.

- $\bar{x}_\perp = (I - U_i U_i^T)\bar{x}_{i+1}$
- $u = \bar{x}_\perp / \|x_\perp\|_2$
- Conduct matrix $\hat{D} = \begin{bmatrix} D_i & U_i^T \bar{x}_{i+1} \\ \bar{0}^T & u^T \bar{x}_{i+1} \end{bmatrix}$, compute the SVD of $\hat{D}$ as the product $\hat{D} = \tilde{U}\tilde{D}\tilde{V}^T$
- Set a threshold ε. All eigenvalues in $\tilde{D}$ which are below the threshold ε are removed. The diagonal matrix composed of the remaining eigenvalues is denoted as D_{i+1}
- Delete from $\tilde{U}$ all the columns that correspond to eigenvalues that were removed above, the matrix composed of the remaining columns in $\tilde{U}$ is denoted as $\breve{U}$
- The updated left eigenspace is calculated as $U_{i+1} = [U_i \mid \bar{u}]\breve{U}$

There are two primary sources of errors in this algorithm: the round off errors incurred in computing the SVD of $\hat{D}$ and the error from truncating the SVD. Standard dense SVD algorithm or fast algorithm suggested by Gu and Eisenstat [48] can be employed to compute the SVD of $\hat{D}$. Both of them are backward stable. The potential instability comes from $\bar{x}_\perp$. If $\bar{x}_\perp$ .is very small, then the computed u may no longer be numerically perpendicular to U_i, leading to serious numerical instability. Hence it is necessary to set a certain threshold to monitor the value of $\|x_\perp\|$. From practice point of view, if $\|x_\perp\|$ is smaller than ε, updating is usually not necessary and can be safely skipped. Applying modified Gram-Schmidt [2] or QR type orthogonalization [2] would be another option to solve this problem thereby making the algorithm numerically robust.

Rounding error is another important error due to truncating the SVD. It can be noticed that U_{i+1} approximates X_{i+1} within an accuracy of δ. Similarly U_i approximates X_i within an accuracy of δ. Therefore it can be concluded that U_{i+1} approximates X_i within an accuracy of 2δ. In general, for $j \le i$, U_i approximates X_j within an accuracy of $(i - j + 1)\delta$. If δ is chosen as ε / N, the approximation of all the images can be secured to an accuracy of ε.

Consider the time complexity of the algorithm, for the data matrix with the size of (mxN), the full fledged SVD from scratch every time would cost $O(mN^3)$. For the proposed algorithm, the computation is spent mostly on computing the SVD of a *broken arrowheaded* matrix $\hat{D}$. If the fast algorithm of Gu and Eisenstat [48] is used, the computation cost at ith updating will be $O(mi)$. Therefore the total computation cost will be $O(mN^2)$. Therefore this algorithm is more computationally efficient, although this will be useful only if N is large enough, which is the common case in the image and video error concealment application.

Before the eigenspace is updated by the current damaged image, the damaged part is considered as missing data and roughly estimated as follows:

Considering adding a vector $\overline{x}_{i+1}$ containing missing elements, partition the incoming vector $\overline{x}_{i+1}$ into $\overline{x}_y$ and $\overline{x}_n$, vectors composed of known and unknown elements respectively and corresponding rows of U_i formed matrices U_y and U_n. The missing values are estimated via the normal equation $\overline{x}_n = U_n U_y{}^{+} \overline{x}_y$; thereby the complete vector bears the fewest standard deviations from the origin with respect to the observations seen so far. (x^+ denotes pseudo-inverse). The $\overline{x}_n$ is derived as follows:

The identity

$$X_{i+1} = \left[X_i \mid \overline{x}_{i+1}\right] = \underbrace{\left[U_i \mid u\right]}_{\hat{U}} \underbrace{\begin{bmatrix} D_i & U_i{}^T \overline{x}_{i+1} \\ \overline{0}^T & u^T \overline{x}_{i+1} \end{bmatrix}}_{\hat{D}} \underbrace{\begin{bmatrix} V_i^T & \overline{0} \\ \overline{0}^T & 1 \end{bmatrix}}_{\hat{V}} \tag{3.19}$$

can be written as:

$$[X_i \mid \overline{x}_{i+1}] = [U_i D_i V_i^T \quad U_i U_i{}^T \overline{x}_{i+1} + uu^T \overline{x}_{i+1}] \tag{3.20}$$

Partition incoming vector $\overline{x}_{i+1}$ into $\overline{x}_y$ and $\overline{x}_n$, vectors composed of known and unknown elements respectively and corresponding rows of U_i form matrix U_y and U_n. If the projection onto residue subspace $uu^T x_{i+1}$, which is small, is omitted and substitute $U_i{}^T x_{i+1}$ as L, we can obtain:

$$\begin{bmatrix} U_y \\ U_n \end{bmatrix} L = \begin{bmatrix} \bar{x}_y \\ \bar{x}_n \end{bmatrix}$$

$$\begin{cases} U_y L = \bar{x}_y \\ U_n L = \bar{x}_n \end{cases} \Rightarrow \bar{x}_n = U_n (U_y)^+ \bar{x}_y \qquad (3.21)$$

Substituting the missing elements in incoming vector with estimated ones followed by the updating scheme will produce the eigenspace which is much closer to the truly one. Missing data estimation definitely will increase the computation complexity. In the worst case, for each updating the complexity will be raised to $O(mi^3)$, but in practice with missing data the pseudo-inverse problem tends to be small, the run time for each updating stay close to $O(mi^3)$ which is dominated by the problem of re-diagonalizing $\hat{D}$.

3.4.3 Performance comparison

Experiments are conducted to illustrate the accuracy and efficiency of the proposed eigenspace updating algorithm. The Oracle research laboratory database [47] was used in these experiments. There are different images of each of 40 distinct subjects with varying lighting, facial expressions, and facial details.

First of all we would like measure the accuracy of proposed incremental algorithm compared with traditional batch mode algorithm. The batch mode algorithm is operated on the matrix containing all images in the training set and performs best in some senses; therefore it serves as the baseline for comparison. The accuracy is measured in terms of the differences between eigenvalues, eigenvectors and reconstruction errors under same remaining energy. The comparison of eigenvalues is shown in Figure 3.13.

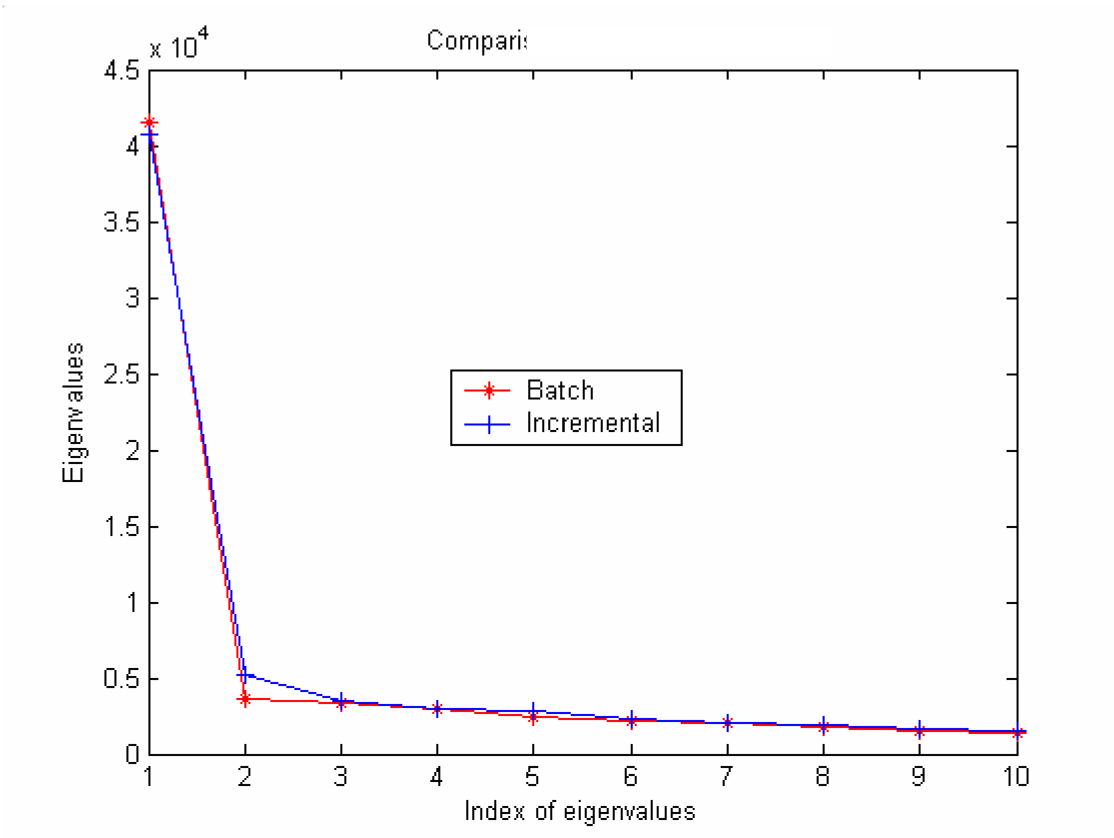


Figure 3.13 Comparison of eigenvalues

Output eigenspaces are visually compared in Figure 3.14. It is observed that eigenfaces produced by batch mode PCA and proposed incremental PCA are visually very similar.

Figure 3.14 The first five eigenvectors of the eigenspace produced by the batch mode PCA (top row) and the corresponding eigenvectors produced by the proposed incremental PCA (bottom row).

Table 3.1 shows the average reconstruction error as a function of the basis dimension, where the basis dimensions are from 10% to 90% of the training set. The average reconstruction error was obtained by transforming images onto the eigenspace represented by the eigenimages and then computing the average reconstruction error per pixel.

Table 3.1 Comparison of reconstruction error

Basis dimension	10%	20%	30%	40%	50%	60%	70%	80%	90%
Reconstruction error of batch	0.2036	0.2030	0.1704	0.1332	0.1192	0.1040	0.0947	0.0853	0.0841
Reconstruction error of incremental	0.2167	0.2140	0.1892	0.1488	0.1205	0.1178	0.0997	0.0903	0.0853

It can be seen from Table 3.1, the reconstruction error decreases when the size of the eigen basis increases. The performance of the incremental algorithm is very close to that of the batch algorithm and difference between the average reconstruction error is quite small. The comparison of reconstruction effect is also illustrated in Figure 3.15

and Figure 3.16. The visual reconstruction quality for the incremental algorithm is comparable with that of the batch algorithm, as the experiment show.

(a) (b)

Figure 3.15 Reconstruction comparisons 1. (a) Batch algorithm with 40% eigen basis, reconstruction error = 0.1442 (b) Incremental algorithm with 40% eigen basis, MSE of reconstruction = 0.1673

(a) (b)

Figure 3.16 Reconstruction comparisons 2. (a) Batch algorithm with 60% eigen basis, reconstruction error= 0.1302 (b) incremental algorithm with 60% eigen basis, MSE of reconstruction = 0.1441

In addition, we also would like to compare the proposed incremental algorithm with the other incremental algorithm. IPCA for classification [51] is chosen as another bench mark. From the angle difference comparison which is shown in table 3.2, we can see the average angular deviation of proposed eigenvectors and batch eigenvectors is much lower than average angular deviation of IPCA-classification eigenvectors and batch eigenvectors. Same trend can be found in comparison of eigenvalues. So the proposed algorithm was also more accurate than the IPCA for classification in terms of eigenvalue and eigenvectors measurements. The latter is more suitable for classification applications.

Table 3.2. Comparison of angular deviation

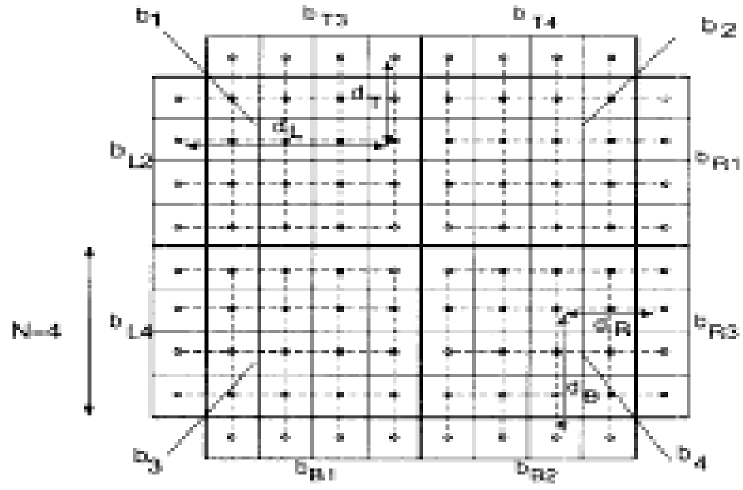
Image set	1	2	3	4	5	6	Avg
Angular deviation (IPCA classification ~batch)	1.0357	1.5083	1.4987	1.4283	1.4604	1.3484	1.4159
Angular deviation (proposed~batch)	6.8293e-015	7.0692e-015	5.2138e-015	7.2006e-015	7.1537e-015	7.4560e-015	6.8204e-15

The proposed algorithm easily updates eigenspace with one new image in an incremental manner without recomputing the basis set from scratch, hence it reduced computation and memory requirements. The performance of the proposed algorithm is comparable to the batch mode algorithm in terms of accuracy and superior to the other incremental algorithm such as IPCA for classification. It is suitable for computing an optimal low dimensional basis in a dynamic environment in which a single new image in continuously included into the image. Therefore it is utilized for the adaptive eigenspace based error concealment.

3.5 Experimental results

The performance of the proposed adaptive eigenspace based error concealment is first tested on decoded video sequence *Peter* [54] with 150 frames and compared with the other two error concealment schemes: fixed eigenspace based error concealment which is described in section 3.2 and interpolation based error concealment. Video sequence *Peter* is mainly composed of a male talking face with multiple appearance variations due to pose changes. The size of each frame is 224x144. Both height and width are multiples of 16 that is the size of one macro block.

Fixed and adaptive eigenspace based error concealment are both learning based error concealment methods. Twenty clean frames from the sequences are trained off line or on line to build the eigenspace model. Fixed eigensapce based error concealment will use this eigenspace model consistently in the remaining 130 frames to repair the error content. Adaptive eigenspace based error concealment will use this eigenspace as initial model and adaptively updates it during the decoding process. How to choose new frames to update eigenspace is described in section 3.3. The incremental PCA updating algorithm with missing data is presented in section 3.4. Both fixed eigenspace based and adaptive eigenspace based error concealment methods are under POCS framework. Two or three iterations of POCS are executed to achieve better convergence results. The interpolation method which is called spatial error concealment in MPEG-2 [32] is used as another bench mark. Its algorithm is shown in Figure 3.17.



$$b_1(i,k) = \frac{d_T b_{L2}(i,N) + d_L b_{T3}(N,k)}{d_L + d_T}$$

$$b_2(i,k) = \frac{d_T b_{R1}(i,1) + d_R b_{T4}(N,k)}{d_R + d_T}$$

$$b_3(i,k) = \frac{d_B b_{L4}(i,N) + d_L b_{B1}(1,k)}{d_L + d_B} \qquad i,k = 1,\cdots,N$$

$$b_4(i,k) = \frac{d_B b_{R3}(i,1) + d_R b_{B2}(1,k)}{d_R + d_B}$$

Figure 3.17 MPEG-2 intra frame error concealment scheme [32]

Each algorithm across different loss rates ranging from 0.02~0.2 is tested. Each algorithm is implemented on 20 damaged frames with the same loss probability under the same quantization level. The average PSNR versus probability is shown in Fig 3.18.

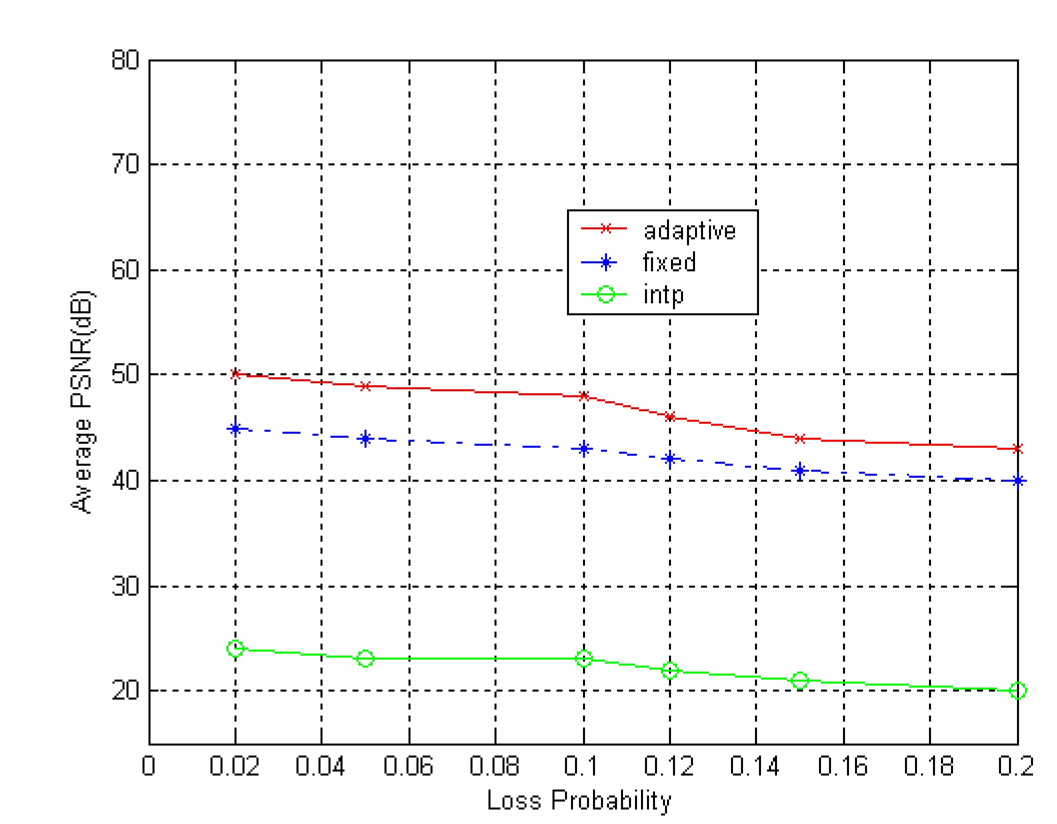


Figure 3.18 Error concealment across different loss rates

It can be seen that the model based error concealement outperforms interpolation based error concealment by up to 25 dB and adaptive eigenspace based error concealment always performs the best. As loss rate increases, the average PSNR of all algorithms decreases. This decrease is more obvious for interpolation based error concealment.

Without errors | Damaged | Interpolation concealed | Fixed eigenspace | Adaptive eigenspace

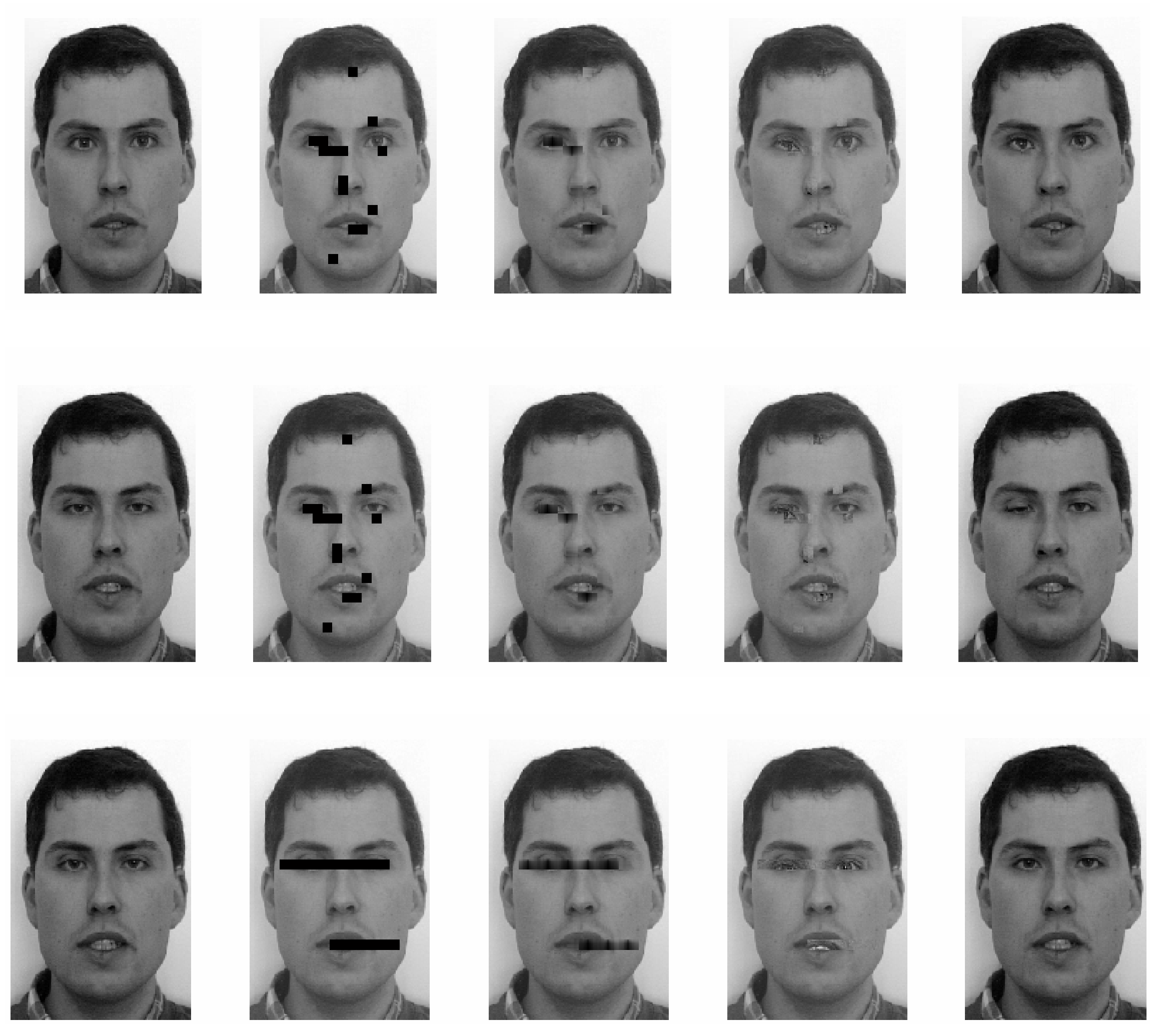


Figure 3.19 Frames 32 (above), 45 (middle) and 50 (bottom) under QP=3, loss rate=5%

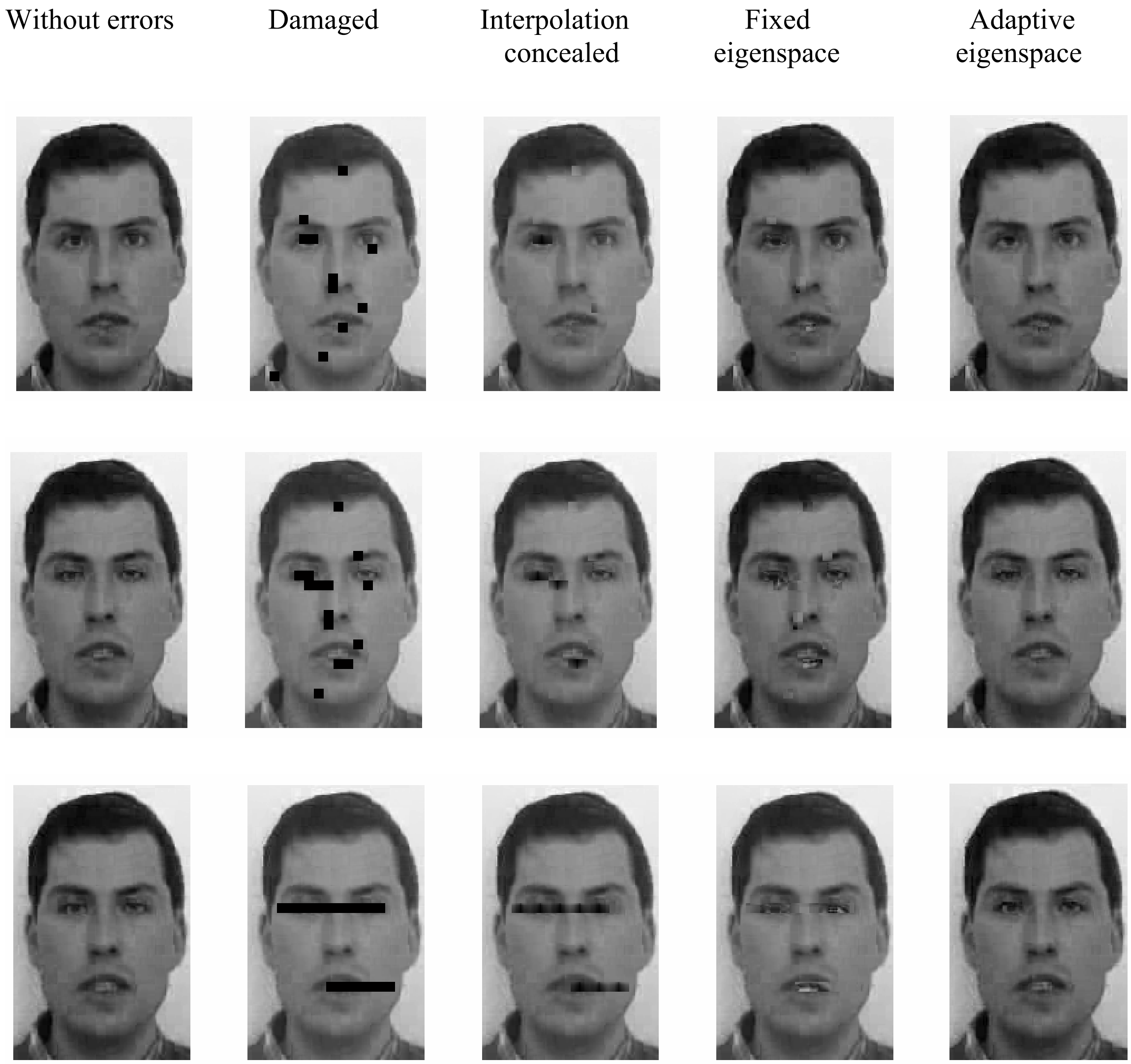


Figure 3.20 Frames 32 (above), 45 (middle) and 50 (bottom) under QP=13, loss rate=5%

The three frames (frames 32, 45 and 50) from the decoded video sequence *Peter* presented in Figures 3.19 and 3.20 visually compare the performances of the three algorithms. Interpolation based error concealment can work well for smooth areas but its performance is poor for burst loss pattern. Fixed eigenspace cannot capture large variations in the sequence; its performance cannot be guaranteed over the entire sequence. When the distance between the damaged image and training set is less than 5 frames, the performance of fixed eigenspace model is quite comparable to adaptive eigenspace model. After that its performance degraded and became worse than the adaptive eigenspace model. When the distance went beyond 10 frames, this degradation trended to slow down. Adaptive eigenspace adapts to the changing appearances. Therefore stable good performances are observed under different loss rates and different loss patterns. Although quantization levels do not affect the objective measurement of performance such as PSNR, the concealed frame has less obvious error concealment artifacts because block artifacts due to coarse quantization overwhelm the imperfection of the error concealment.

The proposed method is also applied to QCIF sequence Miss America. A sample frame from the sequence corresponding to a loss probability of 0.1 is presented. Figures 3.21 and 3.22 illustrate the concealed frame for quantization step QP=1 and QP=13 respectively. Again since the damaged part is not smooth, the interpolation concealed method does not perform well. Adaptive eigenspace based error concealment performs superior to the other two methods. Similar experiments are also conducted on QCIF sequence Carphone and the simulation results are displayed on Figures 3.23 and 3.24.

(a)

(b)

(c)

Figure 3.21 Concealed frames from Miss America (loss rate=0.1, QP=1)
(a) Intepolation concealed
(b) Fixed eigenspace concealed
(c) Adaptive eigenspace concealed

(a)

(b)

(c)

Figure 3.22 Concealed frames from Miss America (loss rate=0.1, QP=13)
(a) Intepolation concealed
(b) Fixed eigenspace concealed
(c) Adaptive eigenspace concealed

(a)

(b)

(c)

Figure 3.23 Concealed frames from Car Phone (loss rate=0.1, QP=1)
(a) Intepolation concealed
(b) Fixed eigenspace concealed
(c) Adaptive eigenspace concealed

(a)

(b)

(c)

Figure 3.25 Concealed frames from Car Phone (loss rate=0.1, QP=13)
(a) Intepolation concealed
(b) Fixed eigenspace concealed
(c) Adaptive eigenspace concealed

3.6 Conclusions

A new model based algorithm is presented in this chapter for error concealment application. The focus is placed on the model based error concealment which can be viewed as second generation error concealment scheme, which is more suitable for the object oriented coding standard such as MPEG-4 [15], where the object has already been extracted and informed in the transmission stream. Since the specifically trained and built eigenspace model captures the video sequence variations more efficiently, better error concealment results are expected compared with heuristic based error concealment. The accuracy of the eigenspace model is the key element to guarantee the satisfactory performance. Therefore a novel adaptive scheme is proposed to build accurate and computationally simple models for this purpose. Current and immediate previous frames are most relevant to eigenspace building, so they are adaptively selected for the eigenspace updating. In this scheme, an incremental PCA with missing data is investigated to update the eigensapce with the received new image vector. Its computational complexity and accuracy are analyzed and evaluated through comparison with batch mode PCA and other incremental PCA. The results of error concealment experiments have shown that the adaptive eigenspace based error concealment has better performance than fixed eigenspace based error concealment and interpolation based error concealment. Also this good performance is stable across different quantization levels, loss patterns and loss rates.

CHAPTER 4

PCA/WAVEFORM HYBRID CODING FOR LOW BIT RATE TRANSMISSION OF FACE SEQUENCES

Video coding is the process of compressing and decompressing a digital video signal. Modern video coding techniques provide the ability to store or transmit the vast amount of data necessary to represent digital images and video in an efficient and robust way. The international standards such as H.261, H.263, and MPEG-2 [10-13] use the state-of-the-art video coding technologies and achieve high compression ratios. The newest coding standard H.264/AVC [19] offers significant improvement in rate-distortion efficiency. Although the dramatic improvements in terms of bit rate reduction have been made, the emerging multimedia applications in which video signals are to be transmitted over internet or mobile networks, where the channel capacity is very limited, address the increasing demand for higher coding efficiency of video coding system. Therefore much research has been carried out on development of very low bit rate video coding technology, such as object-oriented coding and model based coding. [14][15].Due to the important role that face to face video communication plays in near future customer service applications, development of efficient coding methods to represent human faces also become an important problem in the area of image compression for video telephony.

In this chapter a novel video coding system aimed at very low bit rate coding of facial images in video sequences is presented. The very high compression ratio mainly comes from model based coding and more specifically the PCA [1] based approach. Unlike the synthetic 3D model used in MPEG-4 face animation standard, the PCA based model is derived statistically instead of empirically. Conventional waveform based coding is also integrated into the system to enhance the performance of the proposed system by adding generality and robustness. The potential application scenerio of this system is videophone telephony, where a talking head covers the main portion of the image.

The following sections are arranged as follows: Section 4.1 presents the previous work which is related to the proposed research. The concept of model based coding is introduced first, followed by a brief description of MPEG-4 face animation standard [15]. The existing hybrid system [57] which combines model based coding with waveform based coding is also included. In Section 4.2 application of PCA for still image compression and introducing how to make modifications to cater to a video coding system are summarized. The proposed face coding system is introduced in Section 4.3, which includes a detailed discussion of system architecture and technical issues such as compression method, mode selection etc. Simulation results are presented after that. Possibilities for further research in this area are outlined.

4.1 Related works

In this section, related works of model based coding and existing system which investigate the way to combine model based coding with traditional waveform based coding are introduced.

4.1.1 Model based coding

In model based coding [15], a predefined model is known in advance at both encoding and decoding ends. A few parameters of the model can be configured to resemble the object effectively. Therefore instead of transmitting the information of pixel values, only a few parameters of the model need to be encoded and transmitted. If the number of parameters required to fully describe the image is smaller than the raw image size, compression is achieved. Ideally, the model should be developed such that with as few parameters as possible the space of objects to be encoded is covered. In face coding, a model of a face is to be built such that the model covers the possible facial expressions of people.

There are various kinds of model based coding systems. [56][57] Many of them are based on three dimensional model of the object to be encoded. Parameters called "action units" are used to describe how to move the vertices of the three dimensional model to resemble the object to be encoded. The MPEG-4 face animation standard [15] can be classified as belonging to this class. MPEG-4 face animation standard specifies a face in its neutral state as the face model and a number of feature points on this neutral face are used as reference points. A total of 84 feature points, as illustrated in Figure

4.1, are defined. The facial animation parameters are defined by motion of some of these feature points.

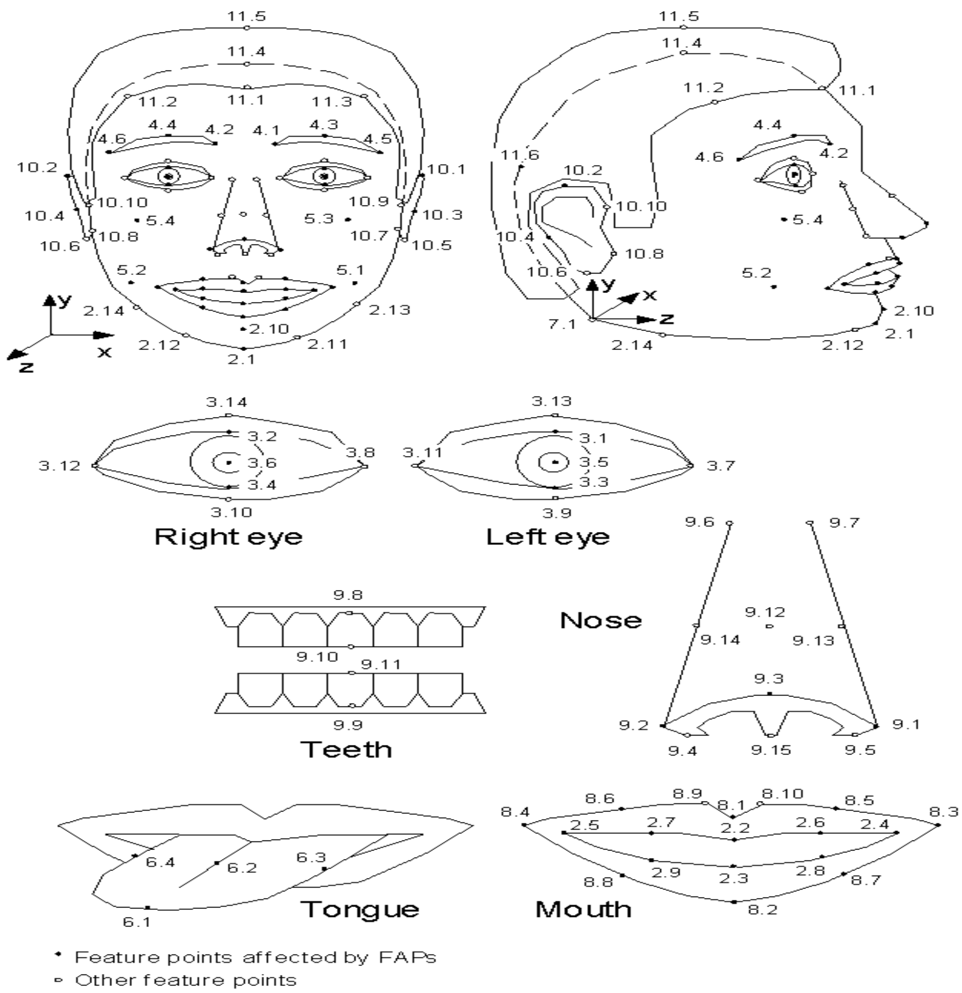


Figure 4.1 MPEG-4 face animation standard feature points [15]

The MPEG-4 standard also defines the facial definition parameter (FDP) which includes an initial shape and texture for the face. Finally, a set of facial animation parameters (FAP) corresponding to a particular facial action are defined. FAP deform a neutral face model according to some specified FAP values. The FAP value for a particular FAP indicates the magnitude of the corresponding action, e.g., a big versus a small smile or deformation of a mouth corner. For an MPEG-4 terminal to interpret the FAP values using its face model, it has to have a predefined model specific animation rules to produce the facial action corresponding to each FAP.

The face model used in the MPEG-4 face animation is built empirically. It is difficult to make the synthesized face model look natural and match the input. Furthermore finding an approach to automatically construct the realistic 3D face model for animation and estimate the parameters of facial motion still remains an open problem today. In this proposed system statistical analysis is used to build a face model from a training set of face images.

4.1.2 Existing model based/waveform based hybrid coding systems

The desire to exploit the advantages of model based coding while maintaining the fidelity of the image by using a reliable fallback coding mode led to the introduction of the switched 3D model based hybrid H.261 coder which was proposed by Chowdhury et al. in 1994. [58] (see Figure 4.2).This type of scheme sets up a switching metric based on the product of the bit-rate and a simple measure of picture quality e.g. PSNR. If the coder is operating in model-based mode and model-based cost is observed to be higher than H.261, then switching occurs at a high level i.e. the entire model-based image is replaced by the H.261 image and vice verse. The mode decision is only made for a complete frame; therefore this scheme can be looked at as a high level switched system where the information from the 3-D model cannot be exploited. During the coding process, the two coding paths, model based coding and H.261

coding, run in parallel, producing two coding streams for each frame. However, only one stream is selected to be transmitted. This is a waste of computational and storage resources.

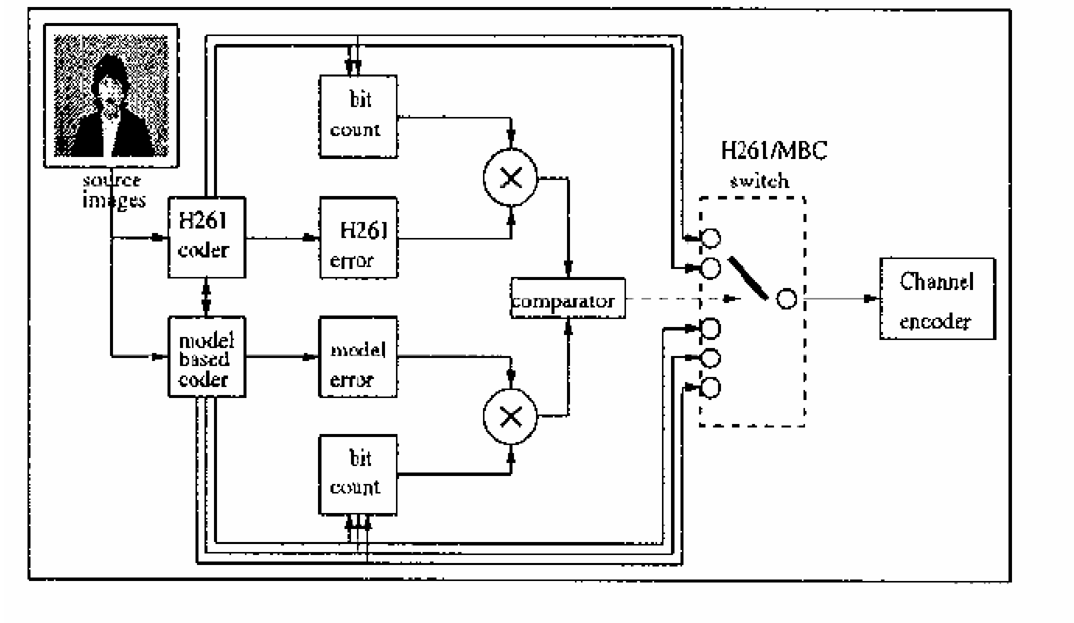


Figure 4.2 System diagram of high level switching [58]

4.2 Principal component analysis for face coding

Although principal component analysis is most widely used for face recognition, substantial work has been done for applying it for image coding. Moghaddam and Pentland [4] presented the eigenface concept for still image coding in a face recognition framework. The same concept can be found in [6] for applications in image coding. In all these papers, simulations have already shown that through the projection of the face to an eigenspace which is previously defined, the still face image can be well represented by very few coefficients; therefore high compression ratio is achieved. All of these coding schemes build the eigenspace by using the complete set of images to be encoded, thus each frame image can be described as a point in the eigenspace. A system with such architecture is suitable for offline use where the image sequence to be compressed is available in advance and can be used to build the eigenspace. The quality of reconstructed image is high, and the performance of coding system is efficient and robust.

In the case of coding moving faces of a video sequence in real time, changes of expression are frequent, and it is impossible to get all the face images to be encoded in the training stage, so the validity of using the eigenspace approach for compressing a face video sequence need to be checked. In this section, some experimental results of applying PCA based face coding for video sequences will be shown. The following assumptions are made:

(1) The camera is static and there is only one talking face, which is the focus of attention in each frame.

(2) Training images coming from database or from video sequences are previously known

(3) The size of the coded image has been previously normalized for principal component analysis purpose.

(4) Prior to coding, the face portion is detected and separated from the background by using some segmentation techniques. From the system point of view, the background can be encoded and transmitted only once at the beginning.

The principle of our face coding technique is similar to that of face recognition, but the approach is modified to cater to video coding application. Basically, it contains two stages: training stage and coding stage. Both of them are described as follows:

Training Stage:

During this stage, first, a set of training images for each person must be obtained. This training set can come from database or directly from video sequence to be coded. The face images in the beginning segment of the video sequences are selected to form the training set. Principal component analysis is performed on the corresponding training set of each person to construct the eigenspace. The eigenspace is known both at the encoder and decoder.

Coding Stage:

Encoding and decoding are included in this stage. At the encoder, the face image to be encoded is transformed into eigenspace. Only M out of 30 significant coefficients corresponding to the M largest eigenvalues are retained, quantized and fixed-length coded. In this experiment, M is set to 5. 8 bit uniform quantizer with quantization step of 200 is applied for coefficient 1, 7 bit uniform quantizer with quantization step of 50 is applied for coefficient 2 and 3, 6 bit uniform quantizer with quantization step of 20 is applied for coefficients 4 and 5. At the decoder, the received coefficients are used to perform linear combination of eigenfaces to get reconstruction.

Experiments have been carried on video sequence *Peter* [54] which is mainly composed of a male talking head. Each frame in this sequence contains 224x144 8 bit-pixels. Both height and width are multiples of 16 that is the size of macroblock for block-based hybrid coding scheme. Frame rate of test sequence is 12.5 frames/s, which implied bit rate of 3.2M bits/s $(224 \times 144 \times 8 \times 12.5)$. 30 frames are selected from the sequence to form the training set, 5 coefficients per frame are retained and transmitted during coding stage. The encoded stream has an average bit-rate of 0.4 kbits/s.

The coding results can be judged by subjective and object evaluations.

Subjective evaluation:

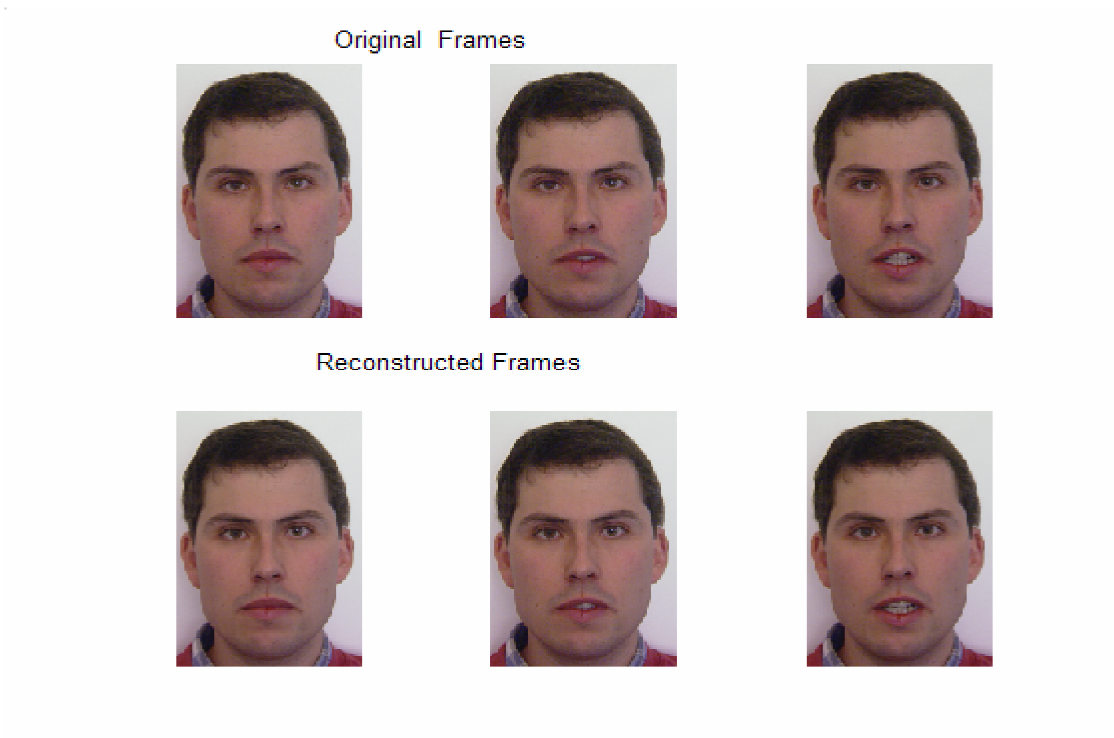


Figure 4.3 Frames with subjectively good quality

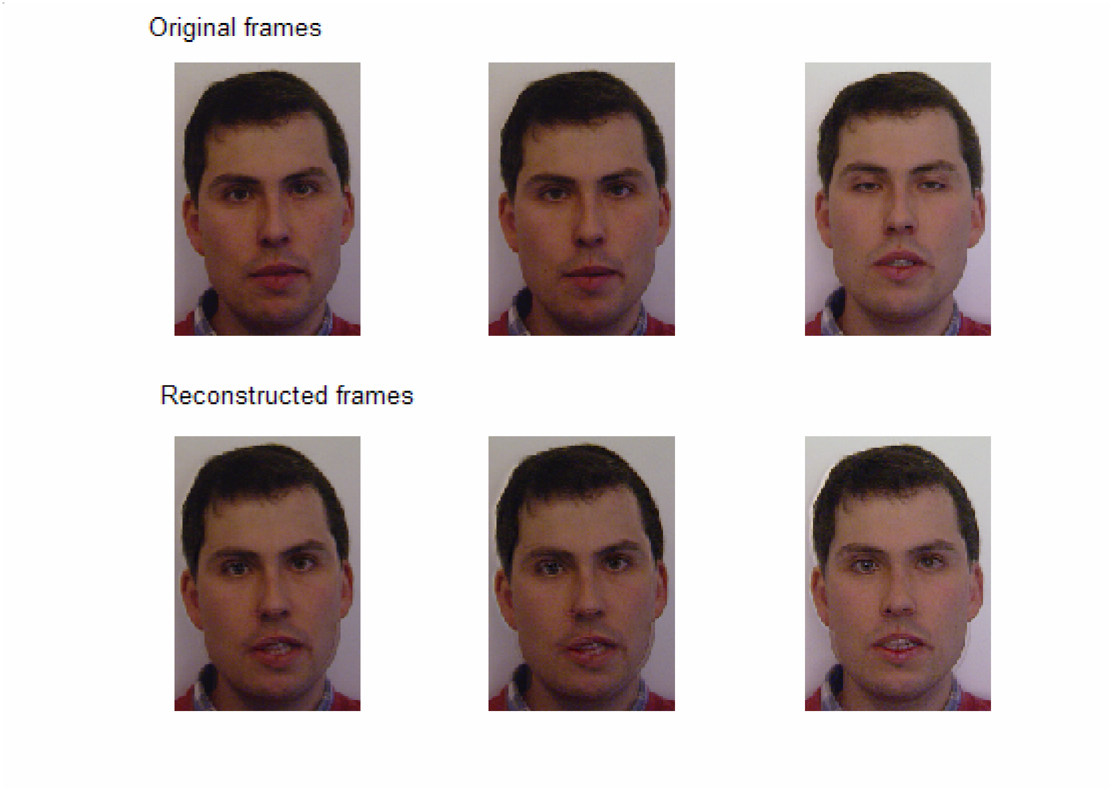


Figure 4.4 Frames with subjectively poor quality

Objective evaluation:

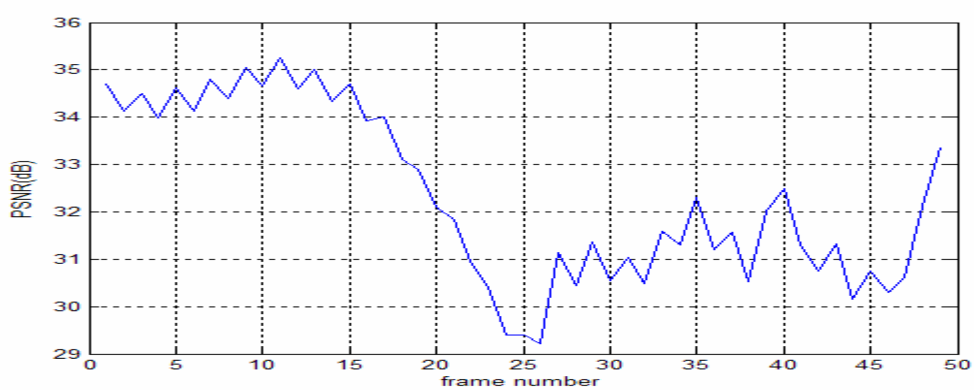


Figure 4.5 PSNR over *Peter* sequence

The coded sequence presented acceptable over all quality based on subjective and objective evaluation. However the quality of the reconstruction varies over the sequence and good quality cannot be granted for every frame. An important change of expression in the face that is not included in the training set will lead the system to poor quality reconstruction.

4.3 Proposed hybrid coding system

Model based compression system can achieve very high compression ratios and high quality reconstruction provided the objects to be coded are within the modeled region. Otherwise the reconstruction quality will be deteriorated. The simulation has already shown that the big face expression change which is not included in training set will lead to poor quality reconstruction. Furthermore, model based compression system is restricted to work on scenes composed of objects that are known by the decoder. In this case, the type of objects to be compressed is limited to talking faces in video sequences.

Compared to model based techniques, conventional waveform based coding techniques are fully automatic and robust. The coding takes place purely on a statistical basis. Therefore it can code arbitrary scene with satisfying reconstruction at the expense of relatively high bandwidth. The transmitted video could be blurred at low bit rates.

To obtain coding efficiency without losing generality, a novel hybrid coding system in which a model based coding and waveform based coding are combined together to complement and support each other is proposed. The model based coding uses principal component analysis based compression scheme and waveform based coding adopts conventional prediction/transform hybrid coding [9]. The system first turns to PCA based coding branch. Once this model works well, high compression ratio will be obtained. If the model based coding does not work well, the failed images can be handled by block-based coding branch. Thus the entire system can work robustly. Furthermore, reconstructed images coming from PCA based coding branch are incorporated into block-based coding to provide secondary reference frames for prediction which will lead to enhanced coding performance. The block diagram of proposed system is shown in Figure 4.6.

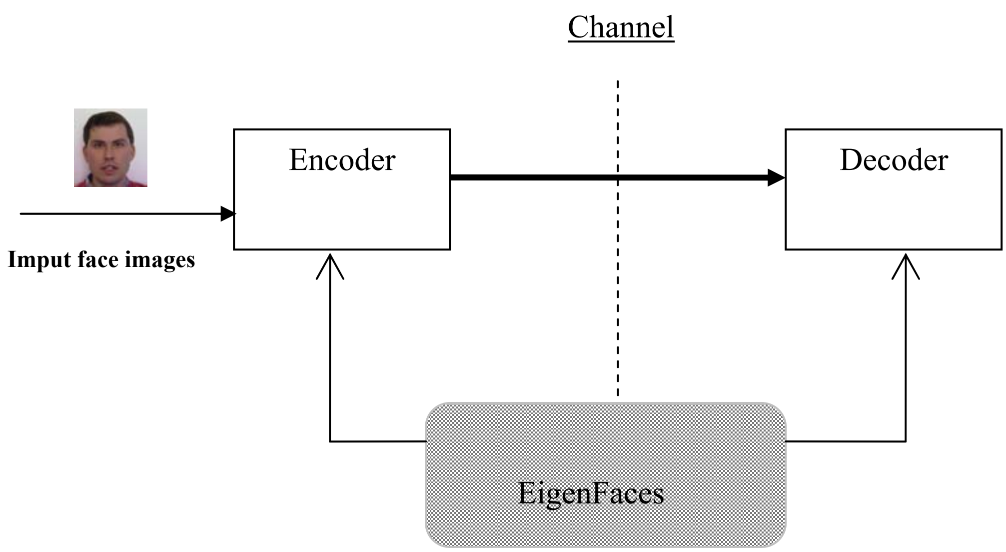


Figure 4.6 The block diagram of proposed coding system

The architecture of the encoder and decoder in this proposed system and technical details such as coding features and rate control scheme will be discussed respectively.

4.3.1 Architecture of encoder and decoder

The whole video coding system includes encoder and decoder. Figure.4.7 shows the architecture of the proposed encoder.

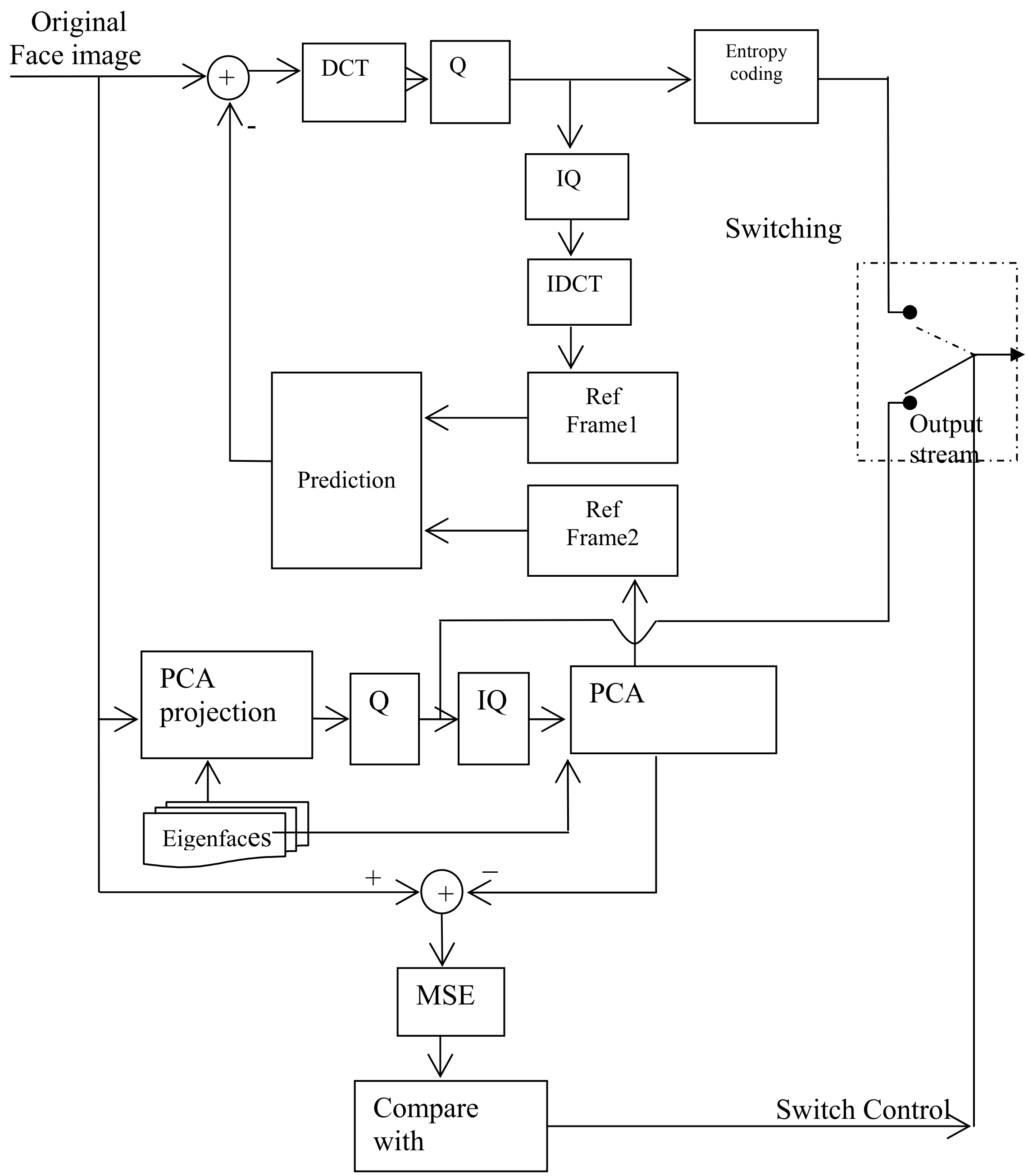


Figure 4.7 Architecture of proposed encoder

The system first turns to PCA based coding branch, the face image to be encoded is projected into eigenspace. Only 5 transform coefficients corresponding to 5 largest eigenvalues are retained, quantized and transmitted. Reconstruction is obtained based on previously known eigenfaces with these 5 left coefficients. According to the reconstruction error which is measured by mean square error (MSE), an additional one-bit flag is sent to indicate what encoder next to do. If the reconstruction error is below a threshold which is set as 24 empirically, "0" is sent and the encoder proceeds to next frame. Otherwise "1" is sent and the system switches to block-based coding path. The PCA reconstructed frame is employed as a secondary reference frame for block-based prediction in addition to the previous reconstructed reference frame. The prediction choice is made based on Lagrangian cost function. If some parts of the image are well approximated by the eigenfaces model, bit rate required for transmission of residue will be reduced.

Block diagram of decoder is shown is Figure 4.8. Model based reconstruction is performed first. Then decoder decides what to do next according to the flag bit. If "0" is received, the model based reconstruction frame is outputted as decoded frame. If "1" is received, the decoder turns to waveform based reconstruction. Since the mode decision and choice of reference frame for prediction are already made at the encoder side, the decoder is simpler than the encoder.

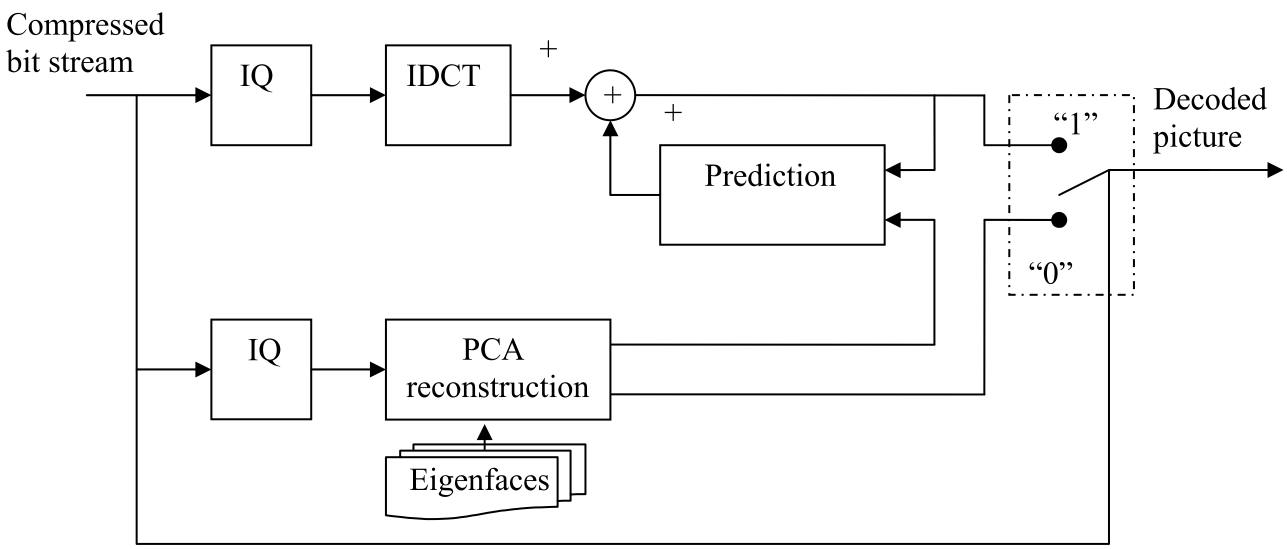


Figure 4.8 Architecture of proposed decoder

4.3.2 Model based coding branch

This coding branch adopts eigenfaces coding approach which is described in the last section. The coefficients are quantized with 3 different uniform quaintizers which are the same as those in the last section.

4.3.3 Block based coding branch

This coding branch adopts conventional prediction/transform block-based hybrid video coding scheme. The whole frame is divided into blocks. The basic processing unit is 16x16 macroblock. After prediction, DCT is applied to the residual information, and then the transform coefficients are scaled and quantized. After that the quantized coefficients are arranged in zigzag scanning order and are compressed by entropy coding. Some features in H.264/AVC [63] standard are applied in this system, including integer 4x4 DCT, multiple frames prediction and CAVLC entropy coding.

Prediction

Prediction is performed at the macroblock level. The first frame of a sequence is "Intra" coded. The predicted value is always from the current PCA based reconstruction. The remaining frames employ prediction either from a previous DCT based reconstruction or from a secondary picture obtained from the current PCA based reconstruction.

Integer DCT and quantization

After the prediction, a transform is applied to decorrelate the data spatially. Like in H.264/AVC [19], the transform in our system is applied to 4x4 blocks, and a separable 4x4 integer transform derived from the DCT is used. Since the inverse transform is defined by exact integer operations, inverse transform mismatches are avoided. The transform is very simple and can be easily implemented using only additions / subtractions and binary shifts. Due to its smaller size, it is not as prone to high frequency "mosquito" artifacts as its predecessors.

The original input images are color images in 4:2:0 format (see Figure 4.9), therefore the residual data within a macroblock are transmitted in the order shown in Figure 4.10. The block labeled "-1" which contains the DC coefficient of each 4x4 luma block is transmitted first, Next, the luma residual blocks 0-15 are transmitted in the order shown (with the DC coefficient set to zero in a 16x16 Intra macroblock). Blocks 16 and 17 contain a 2x2 array of DC coefficients from the Cb and Cr chroma components respectively. Finally, chroma residual blocks 18-25 (with zero DC coefficients) are sent.

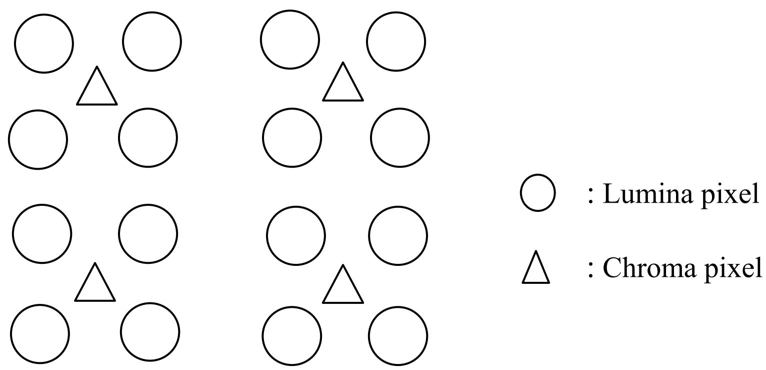


Figure 4.9 4:2:0 sampling format

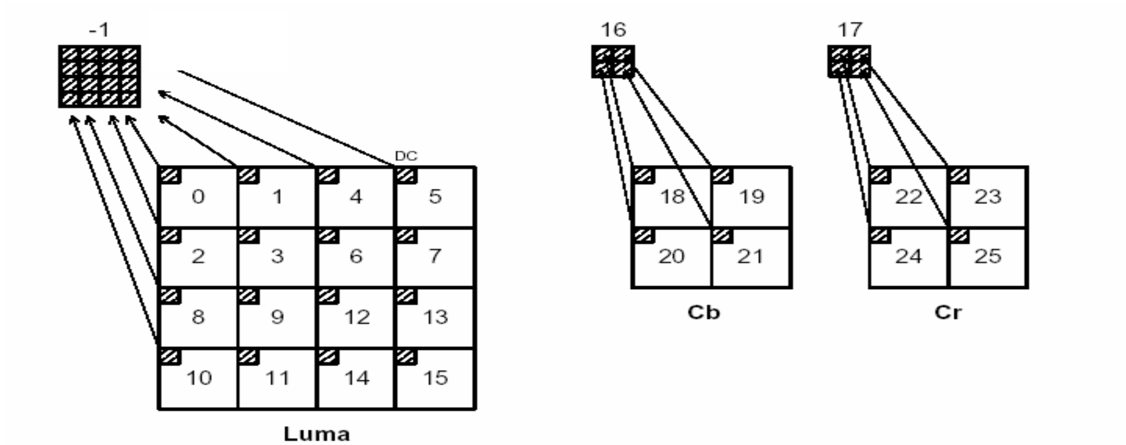


Figure 4.10 Transmission order within one macroblock [18]

The forward transform is defined as

$$Y=(FWF^{T})\otimes E=\left(\begin{bmatrix}1 & 1 & 1 & 1\\ 2 & 1 & -1 & -2\\ 1 & -1 & -1 & 1\\ 1 & -2 & 2 & -1\end{bmatrix}\Bigg[\quad W \quad\Bigg]\begin{bmatrix}1 & 2 & 1 & 1\\ 1 & 1 & -1 & -2\\ 1 & -1 & -1 & 2\\ 1 & -2 & 1 & -1\end{bmatrix}\right)\otimes\begin{bmatrix}a^2 & ab & a^2 & ab\\ ab & b^2 & ab & b^2\\ a^2 & ab & a^2 & ab\\ ab & b^2 & ab & b^2\end{bmatrix}$$

Where a=1/2, b=$\sqrt{\frac{1}{2}}\cos(\frac{\pi}{8})$, c=$\sqrt{\frac{1}{2}}\cos(\frac{3\pi}{8})$, the symbol $\otimes$ indicates that each element of (FWF^T) is multiplied by the scaling factor in the same position in matrix E. While the MB size remains at 16x16, each MB is divided up into 4x4 blocks, and a 4x4 block transform matrix F is applied to every block of pixels. E is a matrix of post scaling factor. This necessary post-scaling step is integrated into quantization.

The inverse transform is given by equation as below:

$$W = F_i^T (Y \otimes E_i) F_i = \begin{bmatrix} 1 & 1 & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & -1 & -1 \\ 1 & -\frac{1}{2} & -1 & 1 \\ 1 & -1 & 1 & -\frac{1}{2} \end{bmatrix} \left(\begin{bmatrix} & & & \\ & & & \\ & Y & & \\ & & & \end{bmatrix} \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \right) \begin{bmatrix} 1 & 1 & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & -1 & -1 \\ 1 & -\frac{1}{2} & -1 & 1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} \end{bmatrix}$$

For the inverse transform, Y is prescaled by appropriate weighting factor from matrix E_i.

The DC coefficients are transformed using a separable 4x4 Hadamard transform with the following matrix:

$$T_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

There are 8 chroma blocks per macroblock. A separable 2x2 transform is used:

$$T_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

For the quantization of transform coefficients, same scalar quantizers as in H.264/AVC are used. The basic forward quantizer operation is as Z_{ij} = round (Y_{ij}/Qstep), where Y_{ij} is a coefficient of the transform. In order to avoid division and/or floating point arithmetic and incorporate the post- and pre-scaling, the definition and implementation are complicated. Total 52 values of step size are supported, indexed by a quantization parameter, QP. Quantization step doubles in size for every increment of 6 in QP.

Scanning

The quantized coefficients are then zigzag read out from the 4x4 coefficient matrix into a single 16 element scan as shown in Figure 4.11. This scan is designed to order the highest-variance coefficients first and to maximize the number of consecutive zero-valued coefficients appearing in the scan.

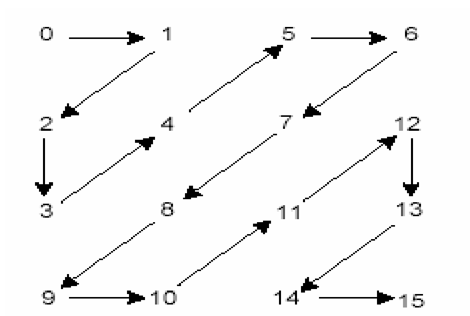


Figure 4.11 Zigzag scan

Entropy coding

CAVLC is used as the entropy coding method. The quantized and zig-zag ordered 4x4 transform block has several favorable properties for CAVLC:

(1) Blocks are often sparse, with long runs of zeros. These zeros are represented compactly by run-length coding in CAVLC.

(2) The non-zero coefficients at the end are often +1/-1. CAVLC codes these “trailing ones” in a compact way.

(3) The number of non-zero coefficients is encoded using a lookup table. Because the number of non-zero coefficients in neighboring blocks is correlated, the choice of table depends on neighboring blocks

(4) The level of nonzero coefficients tends to be larger at the start of the reordered array and smaller toward the higher frequencies. CAVLC takes advantage of this by choosing the VLC look up table for “Level” adaptively according to recently coded level magnitude

The procedure for CAVLC encoding of a block of transform coefficients can be described as below:

(1) Encode the number of coefficients and trailing ones (coeff_token).

The first VLC, coeff_token, encodes both the total number of non-zero coefficients (TotalCoeffs) and the number of trailing +/-1 values (T1). There are 4 choices of look-up table to use for encoding coeff_token. The choice of table depends on a parameter N which can be calculated by the number of non-zero coefficients in upper and left-hand previously coded blocks.

(2) Encode the sign of each T1.

For each T1 (trailing +/-1) signaled by coeff_token, a single bit encodes the sign (0=+, 1=-). These are encoded in reverse order, starting with the highest-frequency.

(3) Encode the levels of the remaining non-zero coefficients.

The level (sign and magnitude) of each remaining non-zero coefficient in the block is encoded in reverse order by looking up 7 VLC tables. The choice of VLC table to encode each level is made based on threshold which is obtained adaptively depending on the magnitude of each successive coded level (context adaptive). In this way, the choice of level is matched to the magnitude of the recently-encoded coefficients.

(4) Encode the total number of zeros before the last coefficient.

TotalZeros is the sum of all zeros preceding the highest non-zero coefficient in the reordered array. This is coded with a VLC. The reason for sending a separate VLC to indicate TotalZeros is that many blocks contain a number of non-zero coefficients at the start of the array and this approach means that zero-runs at the start of the array need not be encoded.

(5) Encode each run of zeros.

The number of zeros preceding each non-zero coefficient (run_before) is encoded in reverse order through look up table. A run_before parameter is encoded for each non-zero coefficient, starting with the highest frequency, with two exceptions:

(a) If there are no more zeros left to encode (i.e. [run_before] = TotalZeros), it is not necessary to encode any more run_before values.

(b) It is not necessary to encode run_before for the final (lowest frequency) non-zero coefficient.

The VLC for each run of zeros can be found by jointly search according to the number of zeros that have not yet been encoded (ZerosLeft) and run_before.

4.3.4 Reference frame selection based on rate-distortion criteria

The choice of video coding algorithm and encoding parameters affect the coded bit rate and the quality of the decoded video sequence. The precise relationship between coding parameters, bit rate and visual quality varies depending on the characteristics of the video sequence.

Rate distortion optimization attempts to maximize image quality subject to transmission bit rate constrains. The trade-off between coded bit rate and image distortion is an example of the general rate-distortion problem in communications engineering. In lossy communication system, the challenge is to achieve target data rate with minimal distortion of the transmitted sequence of images. This problem may be described as follows: "Minimize distortion (D) while maintaining a bit rate R that does not exceed a maximum bit rate R_{max}".

The aim of rate-distortion optimization is to find a set of coding parameters that achieves an operating point as close as possible to the optimum curve. One way to find the optimum solution is by using Lagrangian optimization.

$$D_{REC}(hd, mv, cf) + \lambda_{mode} R_{REC}(hd, mv, cf)$$

Here distortion after reconstruction D_{REC} is measured as the sum of absolute difference (SAD).

Given the displacements for each particular mode, we are computing the overall rate-distortion costs. The distortion is computed by SAD between the reconstructed and original frame, and the rate is computed including the rates of macroblock headers hd, motion parameters mv, and DCT quantization coefficients cf. The mode with the smallest Lagrangian cost is selected for transmission to the decoder. The parameter $\lambda_{\mathrm{mod}\,e}$ is derived from the rate-distortion curve that we computed using H.263 coder. This approach is chosen because of its simplicity and its reproducibility. Following [60], the Lagrange multiplier for the mode decision is chosen as

$$\lambda_{\mathrm{mod}\,e} = 0.85Q^2$$

where Q is the DCT quantization parameter. By computing the Lagrangian cost function, the encoder in proposed system decides which of the two frames is selected as reference frame for prediction for each macroblock.

4.3.5 Simulation results

Experimental results are presented to verify the performance of proposed coding scheme. Experiments are conducted using video sequences 1 and 2 [54], whose frames are mainly occupied by a male talking head. Both sequences contain frames of 224×144 resolution with 12.5 frames/s frame rate. Rate-distortion plots and reconstructed frames are shown for the proposed PCA/waveform hybrid coder and compared with pure H.264 feature like waveform coder. Figure 4.12 shows the rate-distortion curves for sequence 1 and Figure 4.13 shows the rate-distortion for sequence

2. Three decoded frames are depicted in Figure 4.14. The original image is frame 46 of sequence 1. The upper image Fig (4.14.a) corresponds to decoded frame from H.264 feature like coder, while the lower ones Figures 4.14.b and 4.14.c are generated from PCA/waveform hybrid coder.

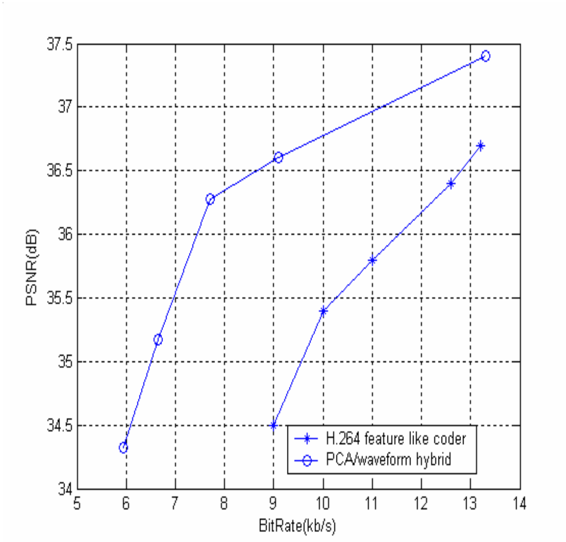


Figure 4.12 Rate-distortion plot for sequence 1

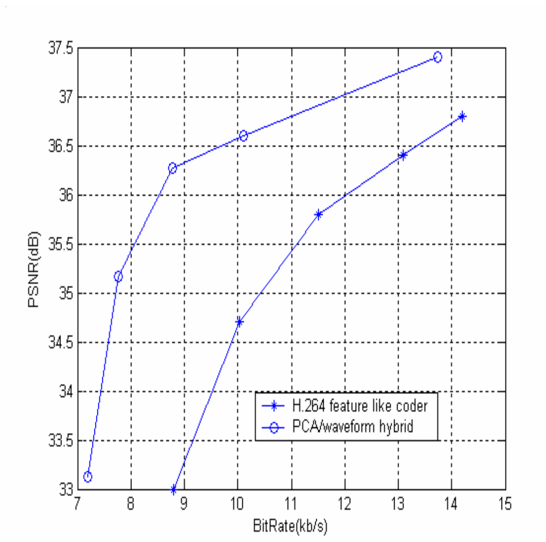
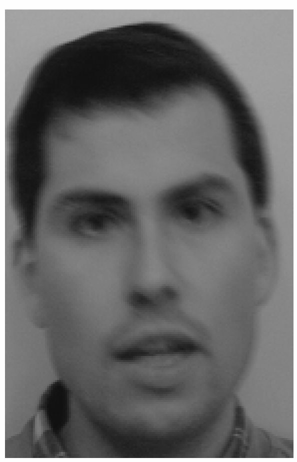
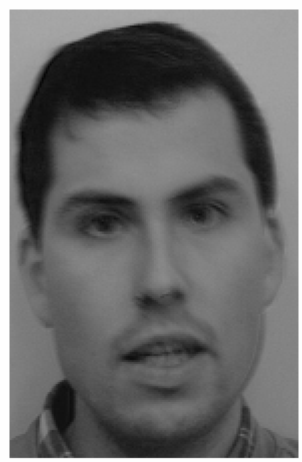


Figure 4.13 Rate-distortion plot for sequence 2

(a)





(b) (c)

Figure 4.14 Comparison of decoded frames
(a)H.264 feature like code (PSNR=33.86dB, 1720bits)
(b)PCA/waveform hybrid coding (PSNR=34.01 dB, 1737 bits, thresholdMSE=17)
(c)PCA/waveform hybrid coding (PSNR=35.4 dB, 102 bits, thresholdMSE=20)

4.4. Conclusions

A novel video coding system for face sequence compression which can be used for video telephony application is proposed. In the proposed system, model based approach is combined with block-based hybrid scheme to code mobile face images efficiently and robustly. Principal component analysis has been proven to be effective and is a widely used algorithm for face recognition. In the proposed system, PCA approach is applied to build facial models composed of eigenfaces and then transform the face image to be encoded into low dimensional eigenspace. Since only a few significant coefficients are encoded and transmitted, the frames which can be coded through the PCA approach can provide large numbers of bit savings for the system. The block based coding in proposed system adopts conventional prediction/transform hybrid architecture. Coding features such as integer transform and CAVLC which are included in the newest H.264/AVC standard [19] are used in the system. Compared to model based coding, block-based hybrid coding is not restricted to certain scene content, but the coding efficiency is relatively low. The advantages of both approaches are combined under rate-distortion optimization control in our system. The reconstructed frame coming from PCA coding path is used as secondary reference frame for prediction in addition to previous reconstructed frame. For each block, the system decides which of the two frames to be selected according to the Lagrangian cost function. The simulation results show that bit rate savings of around 30% are achieved at equal average PSNR compared to pure block-based coding, especially at the low bit rate end.

CHAPTER 5

SUMMARY AND FUTURE WORKS

This research was directed towards applying PCA to build statistical model for error concealment and low bit rate face coding. The objective of this research is to build a non conventional, model based coding scheme which is more suitable for object oriented coding, where the object has already been extracted and indicated in the transmission bit stream. This effort can be viewed as investigating second generation coding scheme to achieve higher compression ratio with high quality, which is required by emerging multimedia applications where more and more bandwidth demanding signals are transmitted over error prone channels. The focus of this research is on the optimization of eigenspace model development in terms of the complexity, accuracy, efficiency and implementation in expected real time applications.

The first contribution of this research is in building a model based frame work for coding and error concealment. The central idea of PCA is to reduce the dimensionality of a data set while retaining as much as possible the variation in the data. Due to this property, a eigenspace model can be built statistically on target object or range of interest (ROI). The engenspace model can capture statistical variation and global information more effectively. Therefore high compression ratio and good error concealment can be expected under such model based frame work.

The second contribution comes from the proposed error concealment method which employs PCA to model the statistical structure of video content in the range of interest (ROI) and use this model as prior knowledge to replenish the lost data. Fixed PCA cannot model the data set with large variations efficiently; hence a novel adaptive PCA scheme is proposed to enhance the accuracy of the eigenspace model on line. The updating is carried out in incremental mode which is suitable for real time applications due to its computational efficiency and low requirement for storage memory. Investigating the incremental updating with the missing data method and its application in the novel adaptive PCA scheme to build accurate and efficient eigenspace model for error concealment are the main goals of this research. Simulations have shown that good error concealment effect can be achieved across different quantization levels, loss patterns and loss rates.

The final contribution lies in the development of a novel video coding system aimed at very low bit rate coding of facial images in video sequences. In the proposed system, model based approach is combined with block-based hybrid scheme to code mobile face images efficiently and robustly. The PCA approach is applied to build a facial model composed of eigenfaces and transform the face image to be encoded into a low dimensional eigenspace. Since only a few significant coefficients are encoded and transmitted, the frames which can be coded through the PCA approach can provide a large number of bit savings for the system. The block based coding in this system adopts conventional prediction/transform hybrid architecture. Coding features such as the integer transform and CAVLC which are included in the newest H.264/AVC standard [19] are used in this system. The advantages of both approaches are combined under rate-distortion optimization control in the system. The reconstructed frame coming from PCA coding path is used as a secondary reference frame for prediction in addition to previous reconstructed frame, which provides further opportunity for compression. The simulation results show that bit rate savings of around 30% are achieved at equal average PSNR compared to pure block-based coding, especially at the low bit rate end.

The generic eigenspace model proposed here is still a very heuristic one and it is difficult to analyze in rigorous fashion without making some assumptions. Adaptive scheme and missing data estimation improve the accuracy and efficiency of the model for a certain object, but still cannot catch up with the large scene changes. Possibilities for improvement include a method for selective echoing of input data, based on perhaps on some characteristics of the previously associated update step, or an analysis of convergence of this method. The eigenspace model can be built statistically, which makes it more flexible to fit the input data and therefore deserves to be further investigated in model based framework which is important for development of future coding scheme.

REFERENCES

[1] K. R. Rao and P. C. Yip, "The transform and data compression handbook", Boca Raton, FL, CRC Press LLC, 2001.

[2] G. Golub and C. Van Loan, "Matrix computation", Third Edition, Baltimore, MD, The Johns Hopkins University Press, 1996.

[3] G. W. Stewart, "Matrix algorithms", vol. I : Basic decomposition, Society for Industrial and Applied Mathematics, Philadelphia, 1998.

[4] M. Turk and A. Pentland, "Eigenfaces for recognition", Journal of Cognitive Neuroscience, vol. 3, pp. 71-86, Jan. 1991.

[5] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation", IEEE Trans. PAMI, vol. 19, pp. 696-710, Jul. 1997.

[6] L. Sirovich and M. Kirby, "Low dimensional procedure for the characterization of human faces", J. of Opt. Soc. Amer. A., vol. 4, pp.519-524, Mar. 1987.

[7] CCIR (ITU-R) 601, "Encoding parameters of digital television for studios", CCIR recommendation 601-2, Vol. XI – part 1, Geneva 1990.

[8] J. R. Jain and A. K. Jain, 'Displacement measurement and its application in interframe image coding", IEEE Trans. Commun., vol.29, pp. 1799-1808, Dec. 1981.

[9] M. Ghanbari, "Standard codecs: image compression to advanced video coding", London, UK, The Institution of Electrical Engineers, 2003.

[10] ITU recommendation H.261, “H.261: video codec for audiovisual services at px64 kbits/s”, Geneva, 1990.

[11] ITU-T Recommendation H.263 Version 2 (H.263+): “H.263+: Video Coding for Low Bit-rate Communication" Jan. 1998.

[12] ISO/IEC 11172-2: video, “MPEG-1: Coding of moving pictures and associated audio for digital storage media at up to about 1.5Mbit/s”, Nov. 1991.

[13] ISO/IEC 13818-2: video, “MPEG-2: Generic coding of moving pictures and associated audio information”, Nov. 1994

[14] ISO/IEC FDIS 14496-2, “MPEG-4: Generic Coding of audio-visual objects: (MPEG-4 video)”, Nov. 1999.

[15] F. Pereira and T. Ebrahimi, “The MPEG-4 Book”, Upper Saddle River, NJ, Prentice Hall, 2002.

[16] B. S. Manjunath, P. Salembier and T. Sikora, “Introduction to MPEG-7”, West Sussex, England, Wiley, 2002.

[17] ISO/IEC JTC1/SC29/WG11/N4301: “Overview of the MPEG-7 standard”. Singapore, Mar. 2001.

[18] I. E. G. Richardson, “H.264 and MPEG-4 video compression”, West Sussex, England, Wiley, 2003.

[19] T. Wiegand et al., “Overview of the H.264/AVC Video Coding Standard”, IEEE Trans. CSVT, vol. 13, pp. 560-576, July 2003.

[20] G. J. Sullivan, P. Topiwala, and A. Luthra, “The H.264/AVC advanced video coding standard: overview and introduction to the fidelity ranged extensions”, SPIE Conference on Applications of Digital Image Processing XXVII vol. 5558, pp.454-474, Aug. 2004.

[21] ISO/IEC JTC1/SC29/WG11/N5231, “MPEG-21 overview v.5”, Shanghai, Oct., 2002.

[22] Y. Wang et al, “Error resilient video coding techniques”, IEEE. Signal Processing Magazine, vol.17, pp.61-82, July 2000.

[23] A. Tamhankar and K. R. Rao, “An overview of H.264/MPEG4 part 10”, 4th EURASIP, vol. 1, pp.1-51, July 2003.

[24] ITU-T Recommendation H.320, “H.320: Narrow band visual telephone systems and terminal requirement”, July 1997.

[25] ITU Recommendation H.221, “H.221: Frame structure for a 64 to 1920 kb/s channel in audiovisual teleservices”, July 1997.

[26] ITU-T recommendation H.323v2, “H.323: Packet based multimedia communications systems”, Mar. 1997.

[27] ITU-T recommendation H.324, “H.324: Terminal for low bit rate multimedia communication”, Feb. 1998.

[28] ITU-T recommendation H. 223, “H.223: Multiplexing protocol for low bit rate multimedia communication”, Mar. 1996.

[29] H. Schulzrinne et al, “RTP: A transport protocol for real time applications”, RFC1889, Jan. 1996.

[30] J. C. Bolot, H. Crepin, and A. Vega-Garcia, “Analysis of audio packet loss in the internet”, 5th workshop on network and operating system support for digital audio and video, pp. 163-174, Durham, Apr. 1995.

[31] S. Hemami and T. Meng, “Transform coded image reconstruction exploiting interblock correlation”, IEEE Trans. IP, vol. 4, pp.1023-1027, July 1995.

[32] S. Aign and K. Fazel, “Temporal and spatial error concealment techniques for hierarchical MPEG-2 video codec”, IEEE Globecom, pp.1778-1783, June 1995.

[33] W. Kwok and H. Sun, “Multi-directional interpolation for spatial error concealment”, IEEE Trans. Consumer Electronics, vol.39, pp.455-460, Aug. 1993.

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

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

[36] W. M. Lam, A. Reibman and B. Liu, “Recovery of lost or erroneously received motion vectors”, IEEE ICASSP, vol. 5, pp. 417-420, April 1993.

[37] J. Zhang, J. F. Arnold and M. R. Frater, ‘A cell-loss concealment technique for MPEG-2 coded video”, IEEE Trans. CSVT, vol. 10, pp.659-665, June 2000.

[38] M. E. Al-Mualla, N. Canagarajah and D. R. Bull, “Error concealment using motion field interpolation”, IEEE ICIP, vol. 3, pp.512-516, Oct. 1998.

[39] D. S. Turaga and T. Chen, “Model-based error concealment for wireless video”, IEEE Trans. CSVT, vol. 12, pp.483-495, June 2002.

[40] J. Huang and T. Chen, "Tracking of multiple faces for human–computer interfaces and virtual environments," IEEE ICME, vol. 3, pp. 1563–1566, Aug. 2000.

[41] D. C. Youla and H. webb, "Image restoration by the method of convex projection: Part 1 – theory", IEEE Trans. Medical Imaging, vol. MI-1, pp.81-94, Oct. 1982.

[42] T. Hastie and W. Stuetzle, "Principal curves", Journal of the American Statistical Association, vol. 84, pp.502-516, June 1989.

[43] T. Cox and M. Cox, "Multidimensional Scaling", London, Chapman and Hall, 1994.

[44] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding", Science, vol. 290, pp.2323-2326, Dec. 2000.

[45] N. Kambhatla and T. K. leen, "Dimension reduction by local principal component analysis", Neural Computation, vol. 9, pp.1493-1516, July 1997.

[46] M. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analyzers", Neural Computation, vol. 11, pp.443-482, Feb. 1999.

[47] Face databse: ftp://ftp.uk.research.att.com/pub/data/orl_faces.zip

[48] M. Gu and S.C. Eisenstat, "A stable and fast algorithm for updating the singular value decomposition", Technical Report YALEU/DCS/RR-966, Yale University, New Haven, CT, 1994.

[49] S. Chandrasekaran et al, "An eigenspace update algorithm for image analysis", Graphical Models and Image Processing, vol. 59, pp.321-332, Sept. 1997.

[50] A. Levy and M. Lindenbaum, "Sequential Karhunen-Loeve basis extraction and its application to images", IEEE Trans. IP, Vol. 9, pp.1371-1374, Aug. 2000.

[51] P. Hall, D. Marshall and R. Martin, "Incremental eigenanalysis for classification", British Machine Vision Conference, Vol.1, pp. 286-295, Sept. 1998.

[52] O. Troyanskaya et al. "Missing value estimation methods for DNA microarrays", Bioinformatics, Vol.17, pp.520-525, 2001.

[53] T. Wiberg, "Computation of principal components when data are missing", Second Symp. Computational Statistics, pp.229-236, 1976.

[54] Face sequence: ftp://ftp.nt.e-technik.uni-erlangen.de/pub/eisert/sequences/

[55] K. R. Rao and J.J. Hwang, "Techniques and Standards for Image Video and Audio Coding", Upper Saddle River, NJ, Prentice Hall, 1996.

[56] M. Kampmann, "Automatic 3-D face model adaptation for model-based coding of videophone sequences", IEEE Trans. CSVT, vol. 12, pp. 172-182, Mar. 2002

[57] J. A.Provine and L. T. Bruton, "3-D model based coding-a very low bit rate coding scheme for video-conferencing", IEEE ISCAS, vol. 2, pp.798-801, May 1996.

[58] M. F. Chowdhury et al., "A switched model-based coder for video signals", IEEE Trans. CSVT, vol. 4, pp. 216-227, June 1994.

[59] Z. Bian and X. Zhang, "Pattern recognition", Beijing, The Tsinghua University Press, pp. 223-227, 1999.

[60] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression", IEEE Signal Processing Mag. Vol. 15, pp. 74-90, Nov. 1998.

[61] L. Torres and D. Prado, “A proposal for high compression of faces in video sequences using adaptive eigenspaces”, IEEE ICIP, pp.I.189-192, Rochester, NY, Sept .2002.

[62] I. E. G. Richardson, “Video Codec Design”, Wiley, 2002.

[63] H.264 reference model software : http://iphome.hhi.de/suehring/tml

[64] Y.Wang, J. Ostemann and Y.Zhang, “ Video processing and communications”, Upper Saddle River, NJ, Prentice Hall, 2002.

[65] R.C. Gonzalez and R.E. Woods, “Digital image processing”, Upper Saddle River, NJ, Prentice Hall, 2002.

[66] K. Sayood, “Introduction to data compression”, San Francisco, CA, Academic Press, 2000.

[67] Y.Wang and Q.Zhu, “Error control and concealment for video communication: a review”, Proceedings of the IEEE, vol. 8, pp. 974-997, May 1998.

[68] H. Lee, D. H. Choi and C. S. Hwang, “Error concealment using affine transform for H. 263 coded video transmissions”, Electronics Letters, vol. 37, pp. 218-220, Feb 2001.

[69] S. Kumar et al. “Error resiliency schemes in H.264/AVC standard”, JVCIR, vol. 17, pp...., 2006.

BIOGRAPHICAL INFORMATION

Xuejun Hu was born in Anhui, China, in 1970. She received her B.S. degree in Electrical Engineering from Anhui University, China in 1992, her M.S. in Electrical Engineering from Shanghai University, China in 1995 and Ph.D degree in Electrical Engineering from University of Texas at Arlington in 2005. From 1995-1998, She was the R&D engineer at institute of image communication and information processing, Shanghai Jiaotong University. Her current research interests focus on image processing, advanced video coding, error concealment and communications.