Model-Based Error Concealment for Wireless Video

Deepak S. Turaga[†] and Tsuhan Chen
Electrical and Computer Engineering
Carnegie Mellon University, Pittsburgh, PA 15213
dturaga@alumni.cmu.edu, tsuhan@cmu.edu

Abstract

We introduce model-based schemes for error concealment of networked video. We build appearance models for specific objects in the scene and use these models to replenish any lost information. Due to the models being designed specific to the object, they are able to capture the statistical variations in the object appearance more effectively, thereby leading to better error concealment performance. We examine statistical modeling techniques from literature and introduce a new efficient and accurate linear model for data representation called the Mixture of Principal Components (MPC), and use these models for error concealment. We simulate lossy network conditions and show that these model-based concealment schemes outperform the traditional concealment schemes across a variety of loss probabilities and bit rates for the coded video.

1. Introduction

Loss of data during transmission of compressed video leads to objectionable visual distortion in the decoded video. In order to minimize distortion at the decoder end, many schemes for error resilience and error concealment have been developed. Error resilience includes schemes at the encoder end where some redundancy is introduced in the bitstream that makes it possible to recover lost information. Examples of these include error correcting codes, data partitioning techniques etc. Error concealment involves post-processing of the video at the decoder end to hide the effect of the transmission errors. Both error resilience and error concealment techniques may be used in conjunction to improve the quality of the decoded video. The focus of this paper is on error concealment techniques, which do not require modification of the coded video.

There is a lot of existing work on error concealment. Most papers on error concealment view it as a post-processing of video, thereby enabling the concealment scheme to be compliant with many different video coding standards, such as H.263, MPEG-1, 2 and 4. Work in this domain includes work by Kwok and Sun [1] and by Aign and Fazel [2], who use spatial domain interpolation. Sun and Kwok [3] also use projections onto convex sets (POCS) for error

[†] The author is currently with Philips Research USA, Briarcliff Manor, NY 10510.

concealment. Chen, Chen and Weng [5] use temporal domain interpolation and overlapped block motion compensation. Lam and Reibman [4] examine error detection and use spatial and temporal interpolation for error concealment. Park, Kim and Lee [6] introduce a concealment scheme that attempts to recover the transform domain coded video coefficients using some constraints on the smoothness of the video pixels. Atzori and De Natale [7] introduce a sketch-based approach to error concealment. Each received frame is decomposed into a set of sketches, and error concealment involves recovering these sketches using some constraints on the sketch continuity, derivatives, etc. Tsekeridou and Pitas [8] introduce concealment schemes for MPEG-2 that use spatial and temporal block matching and interpolation schemes for error concealment. Shirani, Erol and Kossentini [9] use Maximum A Posteriori (MAP) estimates to recover missing binary shape information from MPEG-4 coded sequences.

Some other work on error concealment involves non-standard-compliant coding techniques for the video. Examples of this kind of work include the work by Chou and Chen [10] and by Man, Kossentini and Smith [11]. These papers use wavelet/subband video coders and examine error detection, resynchronization and concealment of errors. There are other papers that focus on error concealment for video over wireless channels. The work by Lu, Letaief and Liou [12] and the work by Zhang, Arnold and Frater [13] are such examples. The work in [12] considers models for the fading process and try to correct for bit errors while the work in [13] considers loss of packets or cells while transmission over wireless networks and examines the concealment of such errors.

The work in this paper also considers concealment of errors caused by packet losses in video. We propose a model-based scheme for error concealment. There has been some work in the past on model based video coding schemes. These include 3-D model based approaches where a 3-D model of the object appearance is built before coding and 2-D model based approaches that use deformable segmentation of the image and affine motion models. A fine overview may be obtained from the papers by Aizawa and Huang [14] and by Pearson [15].

In our proposed error concealment schemes, we build a model for the region of interest, and use this model to replenish any missing information. The region of interest may include the foreground or the background or the different objects in the scene. The model is trained specific to the object of interest and may be trained online, or using some prior information about the object appearance. Such an approach has advantages over generic spatial, temporal and frequency domain interpolation techniques, as the model is created specifically for an object and hence can capture the statistical variations in the object appearance more effectively. The model may also be

adapted to changing object appearance thereby utilizing history information. We believe such model-based error concealment schemes form part of second-generation error concealment techniques.

One requirement of a model-based concealment scheme is the ability to track the object of interest. Hence, such a model based concealment approach is very useful especially for the MPEG-4 standard, which uses object based coding. Since the video bitstream contains information about objects of interest it is easy to build models for them and perform model-based error concealment.

We first examine the principal component analysis (PCA) [16] as a model for our region of interest and find that it is inefficient at capturing data with large amounts of variation. We then examine many non-linear and linear extensions to the PCA for improved efficiency. Among the non-linear extensions is the work by Hastie and Stuetzle [17], who proposed principal surfaces as an alternative to PCA. This involves modeling the data clusters using parameterized surfaces instead of the hyperplanes that PCA uses. Many neural network approximators for these principal surfaces of the high-dimensional data have also been proposed. Among these are the work by Oja [18] and by Kung and Diamantaras [19]. Other non-linear techniques such as Multi-Dimensional Scaling (MDS) [20] have also been introduced. MDS attempts to preserve pair wise distances between data points during the dimensionality reduction so that local relationships are preserved. Recently, other similar approaches to dimensionality reduction such as Locally Linear Embedding (LLE) [21] have also been proposed. LLE also attempts to preserve local relationships between data points during dimensionality reduction. However all these non-linear techniques are computationally intensive and also lack an easy forward-backward transformation, making them unsuitable for our task.

We then examine other linear extensions to the PCA. Among these extensions is the Vector Quantization PCA (VQPCA) [22]. This technique modifies the traditional VQ algorithm by changing the optimization criterion to include reconstruction error. Data samples are partitioned into clusters based on which cluster reconstructs them with smallest error. The parameters of each cluster are then updated using local PCAs and this process is iterated till convergence of parameters. This hard partitioning of data into clusters before dimensionality reduction leads to loss of the global information present in the data. We would like to both exploit the local as well as global information present in the data and so prefer a soft partitioning of the data. The idea of soft partitioning the data while training local PCAs has been examined by Tipping and Bishop [23]. They first introduce an extension to the PCA called the Probabilistic PCA (PPCA) and use a mixture of such PPCAs to represent the data. The problem with their

approach is the fact that the error due to dimensionality reduction is not explicitly minimized, instead the likelihood of observing the data given the model is maximized. This may lead to poor reconstruction performance of the model.

We propose a linear extension to the PCA called *Mixture of Principal Components (MPC)*. Similar to the way that a Gaussian mixture models the data distribution, the MPC automatically models the data using a mixture of eigenspaces. However, instead of optimizing the likelihood of observing the data given the model, the MPC parameters are chosen to minimize the overall reconstruction error. It is efficient, accurate and the reconstruction is easy to compute. We hence use the MPC to model the objects of interest in video sequences and show the improved error concealment performance of such a model-based scheme over using the conventional schemes.

This paper is organized as follows. Section 2 introduces the model-based error concealment scheme. Section 3 includes a discussion of the models to be used for such a model-based error concealment scheme. We introduce the MPC and compare the modeling performance with other linear modeling techniques. The actual derivation for the MPC is included in the Appendix. Section 4 contains results for error concealment using such a model-based concealment scheme. We finally conclude in Section 5 with a summary and directions for future research.

2. Model-Based Error Concealment

In order to conceal any errors in a region of interest, we first build a model for that region of interest. Then the model may be used to replenish any missing data. We highlight this proposed error concealment scheme in Figure 1.

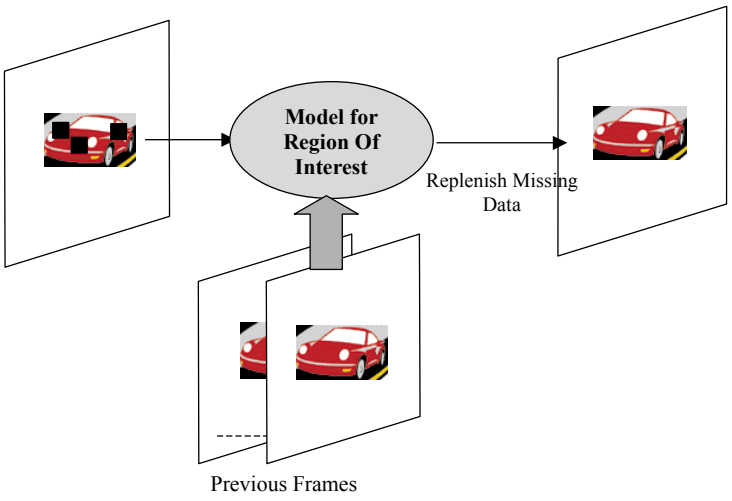


Figure 1. Model-based error concealment scheme

As can be seen from Figure 1, using previously received frames we can build a model for the region of interest, shown as a car in the figure. This model may be trained online, during the transmission of video, or offline, using some prior information about the region of interest. Once we have this model we may use it to correct any errors in the object appearance. For instance, we may project the object with errors onto the model and create a reconstruction, which may then be used to replenish the missing information.

We believe that this error concealment scheme is part of second-generation error concealment techniques, due to this object-based notion. Such an object oriented error concealment scheme has some advantages over previously proposed concealment schemes. Firstly, by tuning the model to a specific object we can capture the statistical variations in the object appearance more effectively, as opposed to using a generic interpolation scheme. Secondly, the model may also be updated online in order to use all the relevant history information. Both of these advantages lead to better error concealment.

As may be seen from Figure 1, this process of error concealment consists of two stages, first the projection onto the model to obtain the reconstruction for the object, and secondly replacing the missing data using this reconstruction. Both these stages may be viewed as projections onto convex sets. The models that we use in this paper involve linear combinations of a set of basis vectors to reconstruct the object, and hence the set of all such reconstructions is convex. Also, the set of all objects or regions of interest with all values same, except for some specific errors, is also convex. Given that both our operations are thus projections onto convex sets, we may iterate the process to obtain better results. This technique of iterative projections onto convex sets

(POCS) was proposed for image restoration by Sezan and Tekalp [24]. Our error concealment scheme may be summarized as follows.

1. *Project region of interest onto the model to obtain reconstruction.*
2. *Replace missing data in region of interest using reconstruction.*
3. *Iterate 1 and 2 until convergence*

This approach is very useful especially for the MPEG-4 standard, which uses object based coding, thereby making it easy to determine regions of interest and build appropriate models for them. In the absence of such object-based information, a tracker for the object of interest may be used to determine the object location. For instance, if the object of interest is the face of a person, a face tracker may be used to locate it.

In order to perform model-based error concealment, the modeling technique has to satisfy some requirements. The model should be able to capture the statistical variations in the object appearance accurately and efficiently. Simultaneously it should not require computationally expensive operations. Efficiency and low complexity are as critical as accuracy as such error concealment schemes are applied at the decoding end, where there are severe constraints on available memory and computation power. Hence we introduce a model that involves linear operations for efficient dimensionality reduction, while attempting to capture the object properties as accurately as possible. The specifics of the modeling technique, for this task of error concealment, are described in Section 3.

3. Model Description

As mentioned in Section 2, we require an accurate and efficient model with low computation complexity for error concealment. We thus examine dimensionality reduction techniques, as they attempt to model a data set as efficiently as possible. There are many linear and non-linear techniques that have been proposed in literature to solve this dimensionality reduction problem.

Among the linear dimensionality reduction techniques is the Principal Components Analysis (PCA). Given a set of objects or data vectors, PCA identifies the principal directions of variation in the data space. These principal directions of variation correspond to the eigenvectors of the covariance matrix of the data and may be used to represent the data. These eigenvectors may be ranked in the order of importance based on the magnitude of their corresponding eigenvalues, with the eigenvector corresponding to the largest eigenvalue being the most significant one and so on. These eigenvectors are used as a basis set to reconstruct the original data vectors. Since we know the principal directions, we may discard some eigenvectors,

corresponding to small eigenvalues without incurring a great increase in reconstruction error. This allows us to model the data using a small set of eigenvectors. As mentioned before, the PCA is inefficient at capturing large variations in the data set, e.g., if the data comes from multiple clusters. Non-linear modeling techniques provide greater efficiency and accuracy for such data sets. However non-linear modeling techniques have problems such as high computation complexity and the absence of a simple forward backward transformation between the low-dimensional representation and the actual data set.

We would like to have the advantages of using a linear modeling technique, however we would like to extend the PCA to improve the modeling performance for data with large variations. Hence we propose a new model called ***mixture of principal components (MPC)*** to represent the data instead of using a single PCA. This is analogous to capturing the probability density of data using a mixture of Gaussians instead of using a single Gaussian. The MPC is designed to optimize the reconstruction error, however we avoid hard partitioning of the data in order to exploit both the local as well as the global information.

As an illustration of data belonging to multiple clusters we collect a face sequence with the person moving his head from left to right, thereby showing broadly three poses, left, center and right. We cropped the faces in a 32×32 window using a face tracker developed by Huang and Chen [25] to provide the location of the faces. We determine the first three eigenvectors of the data using PCA and plot the corresponding coefficients in 3-D space. These are as shown in Figure 2.

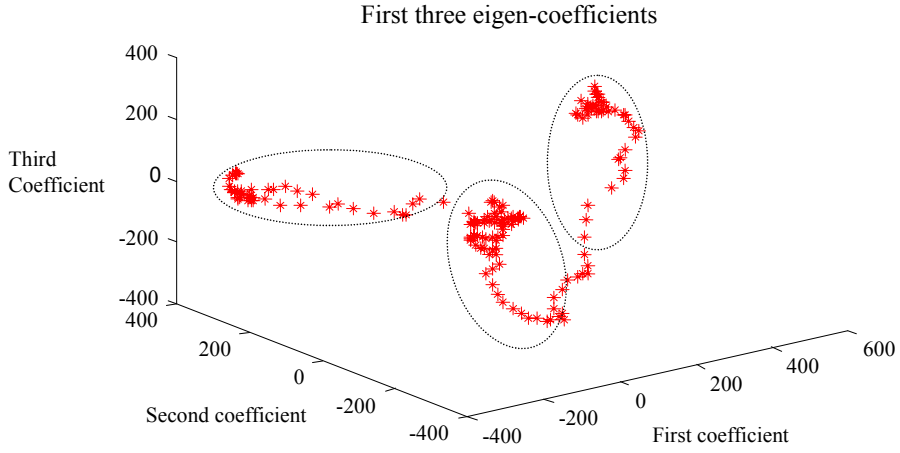


Figure 2. First three eigen-coefficients of real face data

From the plot we can see that the data may be approximated using three clusters, one each corresponding to the left, the right and the center poses. It is clear that to model such data a mixture of Gaussians, or a mixture of eigenspaces is more efficient than using a single Gaussian or a single eigenspace.

As mentioned in Section 1, there are other linear extensions to the PCA such as the Mixture of PPCA and the VQPCA. All of these extensions, including the MPC, can be classified into those that use hard or soft clustering techniques and those that are optimized in terms of the squared error or the likelihood of observing the data. We group these appropriately in Table 1.

Table 1. Linear Extensions to the PCA

	Optimize Likelihood of Observing Data	**Optimize Reconstruction Error**
Hard Clustering		VQPCA
Soft Clustering	Mixture of PPCA	MPC

As may be seen from Table 1, Mixture of PPCA uses soft clustering of data while optimizing the likelihood of observing data, while VQPCA uses hard clustering and optimizes the reconstruction error. The MPC uses soft clustering of the data while trying to optimize the reconstruction error. It is known from literature [23] that the reconstruction error performance of the VQPCA is better than the reconstruction error performance of the Mixture of PPCA. This is to be expected, as the Mixture of PPCA is not designed to optimize the reconstruction error. The focus of this paper is on optimizing the reconstruction error and the MPC is designed to minimize the reconstruction error for a set of training data. Hence we also include a brief comparison of the MPC, in terms of reconstruction error performance, with the VQPCA.

This section is organized as follows. We first describe the notation used in this paper to describe the MPC parameters. We then describe the reconstruction task to identify the optimization criterion. We then describe how the parameters for the MPC may be trained using a set of training data, while optimizing the criterion identified.

3.1. Notation

N : Number of training data vectors
D : Dimension of data vectors
M : Number of mixture components
P : Number of eigenvectors in each mixture component
$\mathbf{x}_i$: Training Data vectors $(i = 1 \ldots N)$
$\hat{\mathbf{x}}_{ij}$: Reconstruction for training vector i from mixture component j $(i = 1 \ldots N, j = 1 \ldots M)$

$\hat{\mathbf{X}}_i$: Matrix with M training vector reconstructions ($\hat{\mathbf{x}}_{ij}$) as columns ($D \times M$)

$\mathbf{y}_i$: Test vector i

$\hat{\mathbf{y}}_{ij}$: Reconstruction for test vector i from mixture component j ($j = 1 \ldots M$)

$\hat{\mathbf{Y}}_i$: Matrix with M test vector reconstructions ($\hat{\mathbf{y}}_{ij}$) as columns ($D \times M$)

$\mathbf{m}_j$: Mean of the j^{th} mixture component

$\mathbf{u}_{jk}$: k^{th} eigenvector of j^{th} mixture component ($j = 1 \ldots M, k = 1 \ldots P$)

$\mathbf{U}_j$: Eigenvector matrix for j^{th} mixture component ($D \times P$)

$\mathbf{w}_i$: Weight vector to combine M reconstructions for vector i. It has elements w_{ij} ($j = 1 \ldots M$)

3.2. Reconstruction Task

Our approach to reconstruction consists of linearly combining individual reconstructions from a mixture of component eigenspaces. Given a data test vector $\mathbf{y}_i$, we first project it onto each of the component eigenspaces to obtain individual reconstructions $\hat{\mathbf{y}}_{ij}$ and then linearly combine these individual reconstructions to obtain the representation that is closest to the original data vector $\mathbf{y}_i$. The individual reconstruction $\hat{\mathbf{y}}_{ij}$ for test vector i from mixture component j is obtained as shown in the following equation.

$$\hat{\mathbf{y}}_{ij} = \mathbf{m}_j + \sum_{k=1}^{P} \left[\left(\mathbf{y}_i - \mathbf{m}_j \right)^T \mathbf{u}_{jk} \right] \mathbf{u}_{jk}$$

These individual reconstructions are then linearly combined using a set of weights. We show an illustration of our approach in Figure 3.

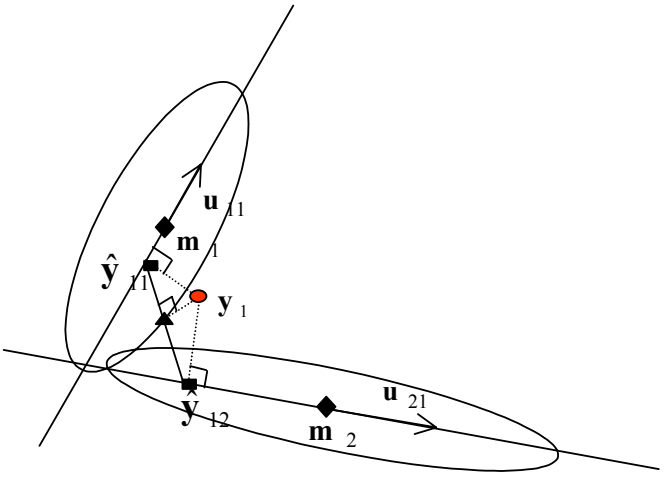


Figure 3. Illustration of mixture of eigenspaces

In Figure 3 we show data reconstruction using a mixture of two component eigenspaces, each with one eigenvector. The component eigenspaces have means $\mathbf{m}_1$ and $\mathbf{m}_2$, and eigenvectors $\mathbf{u}_{11}$ and $\mathbf{u}_{21}$ respectively. In the figure, the means are shown as black diamonds and the direction of the eigenvectors is shown as a line passing through the corresponding means. Given a data sample $\mathbf{y}_1$, shown as a dark circle, we first project it onto each of the component eigenspaces to obtain $\hat{\mathbf{y}}_{11}$ and $\hat{\mathbf{y}}_{12}$. We then linearly combine these two projections to obtain the best reconstruction for the data, shown as the dark triangle in the figure. Due to the nature of the linear weighting, the best combination lies along the line joining the two individual reconstructions. The weights are chosen so that the resulting combination is as close to the data sample, i.e., it lies on the perpendicular from the data sample to the line joining the two individual reconstructions.The weights are solved for individually for each of the test vectors. The only constraint that we impose on the weights is that they are required to sum to one. The weights are chosen by optimizing the criterion shown in equation (1).

$$\min_{\mathbf{w}_i} \left\| \mathbf{y}_i - \sum_{j=1}^{M} w_{ij} \hat{\mathbf{y}}_{ij} \right\|^2 = \min_{\mathbf{w}_i} \left\| \mathbf{y}_i - \hat{\mathbf{Y}}_i \mathbf{w}_i \right\|^2 \tag{1}$$

The matrix $\hat{\mathbf{Y}}_i$ has the individual reconstructions $\hat{\mathbf{y}}_{ij}$ as columns and the weight vector $\mathbf{w}_i$ has the individual weights w_{ij} as entries. During this optimization, we need to add in the constraint on the weights using the Lagrange multiplier. Hence equation (1) may be rewritten as below.

$$\min_{\mathbf{w}_i, \lambda} \left[\left(\mathbf{y}_i - \hat{\mathbf{Y}}_i \mathbf{w}_i \right)^T \left(\mathbf{y}_i - \hat{\mathbf{Y}}_i \mathbf{w}_i \right) + \lambda \left(\mathbf{w}_i^T \mathbf{1} - 1 \right) \right] \text{ where } \mathbf{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} (M \times 1) \tag{2}$$

Taking derivatives with respect to $\mathbf{w}_i$ and λ and setting the result to zero we get the following equations for the weights.

$$\begin{aligned} 2\hat{\mathbf{Y}}_i^T \hat{\mathbf{Y}}_i \mathbf{w}_i - 2\hat{\mathbf{Y}}_i^T \mathbf{y}_i + \lambda \mathbf{1} &= 0 \\ \mathbf{w}_i^T \mathbf{1} = \mathbf{1}^T \mathbf{w}_i &= 1 \end{aligned} \tag{3}$$

The two equations in (3) may be grouped together as follows

$$\begin{bmatrix} 2\hat{\mathbf{Y}}_i^T \hat{\mathbf{Y}}_i & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w}_i \\ \lambda \end{bmatrix} = \begin{bmatrix} 2\hat{\mathbf{Y}}_i^T \mathbf{x}_i \\ 1 \end{bmatrix} \tag{4}$$

We can thus solve for the weights using equation (4).

3.3. Training of the Mixture Parameters

The previous section describes the reconstruction of data given the mixture parameters, i.e., the means and eigenvectors for each component eigenspace. This section focuses on determining these mixture parameters given a set of training data. Given a set of data vectors with variations, we would like to automatically train a mixture of eigenspaces so that the reconstruction error is as small is possible. We would like to train these parameters of the component eigenspaces automatically. We formulate this problem of training as a minimum error optimization problem and provide a solution using an iterative Expectation Maximization (EM) kind of algorithm. We iteratively update the means and the eigenvectors, one by one until a converged result is obtained.

Given a set of N training data vectors, we model them using an MPC containing M eigenspaces, each of which has P eigenvectors. We need to find the means and the eigenvectors for each of these mixture components so that the training data is reconstructed with as small an error as possible. This problem of minimizing the squared error through the choice of the means and the sets of eigenvectors may be mathematically written as in equation (5).

$$\min_{\mathbf{m}_j, \mathbf{u}_{jk}} \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{x}_i - \sum_{j=1}^{M} w_{ij} \left[\underbrace{\mathbf{m}_j + \sum_{k=1}^{P} \left[\left(\mathbf{x}_i - \mathbf{m}_j \right)^T \mathbf{u}_{jk} \right] \mathbf{u}_{jk}}_{\hat{\mathbf{x}}_{ij}} \right] \right\|^2 \tag{5}$$

It may be seen from equation (5) that we use the same scheme to reconstruct the training data as described in Section 3.2. The weights are recomputed for each of the training data vectors, and are used to update the means and the eigenvectors, but are not part of the model. The method for the weight computation is as described in Section 3.2.

The approach we adopt to solving this optimization problem is an iterative one, similar to the EM algorithm for Gaussian mixture training. We first initialize the means and the eigenvectors for the different components randomly. We compute the weights for each data vector following which we update the means, while keeping the eigenvectors fixed. We then use the new means to update the set of eigenvectors. After this we recompute the weights and repeat the update procedure till convergence.

3.3.1. Solutions for the means

The solution for the means is obtained by taking derivatives of the optimization criterion and setting the result to zero. During this process the eigenvectors are kept fixed. The resulting solution for the mean of mixture component q is as shown in equation (6).

$$\mathbf{m}_q = \frac{1}{\sum_{i=1}^{N} w_{iq}^2} \left[\sum_{i=1}^{N} w_{iq} \left(\mathbf{x}_i - \sum_{j=1, j \neq q}^{M} w_{ij} \hat{\mathbf{x}}_{ij} \right) \right] \tag{6}$$

The derivation for this solution is included in the Appendix. From the equation we can see that the mean for a component is basically the weighted sampled mean of the data, with reconstructions from other component eigenspaces removed first.

3.3.2. Solution for eigenvectors

We may now fix means and use these new values to update the eigenvectors. The solution for the eigenvectors for a mixture component eigenspace may be expressed as an eigenvalue-eigenvector problem. The eigenvectors for the r-th mixture component may be obtained as eigenvectors of the matrix $\mathbf{C}_r$ that is defined in equation (7).

$$\mathbf{C}_r = \frac{1}{N} \sum_{i=1}^{N} \begin{bmatrix} w_{ir} \left[(\mathbf{x}_i - \mathbf{m}_r) \mathbf{x}_i^T + \mathbf{x}_i (\mathbf{x}_i - \mathbf{m}_r)^T \right] - \\ \sum_{j=1}^{M} w_{ij} w_{ir} \left[(\mathbf{x}_i - \mathbf{m}_r) \mathbf{m}_j^T + \mathbf{m}_j (\mathbf{x}_i - \mathbf{m}_r)^T \right] - \\ \sum_{j=1, j \neq r}^{M} w_{ij} w_{ir} \sum_{k=1}^{P} \left[\mathbf{u}_{jk}^T (\mathbf{x}_i - \mathbf{m}_j) \right] \left[(\mathbf{x}_i - \mathbf{m}_r) \mathbf{u}_{jk}^T + \mathbf{u}_{jk} (\mathbf{x}_i - \mathbf{m}_r)^T \right] - \\ w_{ir}^2 (\mathbf{x}_i - \mathbf{m}_r) (\mathbf{x}_i - \mathbf{m}_r)^T \end{bmatrix} \tag{7}$$

The first P eigenvectors of this matrix are the desired eigenvectors of the mixture component. After updating the eigenvectors, we now recomputed the weights for each of the training data vectors and repeat the update process for the means and eigenvectors till convergence.

3.3.3. Special Case for M=*1*

As an illustration we consider the case when the number of mixture components is one, when our solution should collapse to the PCA. The mean may be obtained as in equation (6). With only one component, we can make a lot of simplifications.

$$\mathbf{m}_q = \frac{1}{\cancel{\sum_{i=1}^{N} w_{iq}^2}\ N}\left[\sum_{i=1}^{N} \cancel{w_{iq}}\left(\mathbf{x}_i - \cancel{\sum_{j=1, j\neq q}^{M} w_{ij}\mathbf{x}_{ij}}\right)\right]$$

All the weights are set to one, since there is only one component, and the reconstructions from other mixture components may be removed. The solution may be rewritten as in equation (8).

$$\mathbf{m}_q = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \tag{8}$$

This is clearly the sampled mean, which is identical to the PCA. After examining the mean, we now examine the matrix $\mathbf{C}_r$ defined as in equation (7).

$$\mathbf{C}_r = \frac{1}{N}\sum_{i=1}^{N}\left[\begin{array}{l} \cancel{w_{ir}}\left[(\mathbf{x}_i - \mathbf{m}_r)\mathbf{x}_i^T + \mathbf{x}_i(\mathbf{x}_i - \mathbf{m}_r)^T\right] - \\ \cancel{\sum_{j=1}^{M} w_{ij}w_{ir}}\left[(\mathbf{x}_i - \mathbf{m}_r)\mathbf{m}_j^T + \mathbf{m}_j(\mathbf{x}_i - \mathbf{m}_r)^T\right] - \\ \cancel{\sum_{j=1, j\neq r}^{M} w_{ij}w_{ir}\sum_{k=1}^{P}\left[\mathbf{u}_{jk}^T(\mathbf{x}_i - \mathbf{m}_j)\right]\left[(\mathbf{x}_i - \mathbf{m}_r)\mathbf{u}_{jk}^T + \mathbf{u}_{jk}(\mathbf{x}_i - \mathbf{m}_r)^T\right]} \\ \cancel{w_{ir}^2}(\mathbf{x}_i - \mathbf{m}_r)(\mathbf{x}_i - \mathbf{m}_r)^T \end{array}\right]$$

Since there is only one mixture component, the third term vanishes, all the weights are set to one and the summation over the number of mixtures may be replaced with the single term with $j = r = 1$. Using this simplification, the matrix $\mathbf{C}_r$ may be rewritten as below.

$$\mathbf{C}_r = \frac{1}{N}\sum_{i=1}^{N}\left[(\mathbf{x}_i - \mathbf{m}_r)(\mathbf{x}_i - \mathbf{m}_r)^T\right] \tag{9}$$

Clearly, this is identical to the sample covariance matrix for the PCA.

3.4. Simulation Results

In order to evaluate the performance of the MPC we create some sample test data. One issue that needs to be addressed before modeling the data is the choice of the number of mixture components *M* and the number of eigenvectors *P* per component. Currently we do not have an analytical solution to determine these parameters, and they are determined empirically. The number of mixture components *M* is incremented while measuring the performance of the model in terms of the data representation error, until we realize that the improvement in performance is

marginal. For any given M, the number of eigenvectors P is chosen so that we capture 90% of the energy of the original data.

We create a set of 3-D data, distributed in two clusters and model the data using two mixture components with one eigenvector each. We create 3-D data clusters as they are easy to visualize and may be shown on a plot. The resulting parameters for the MPC after training are shown in Figure 4.

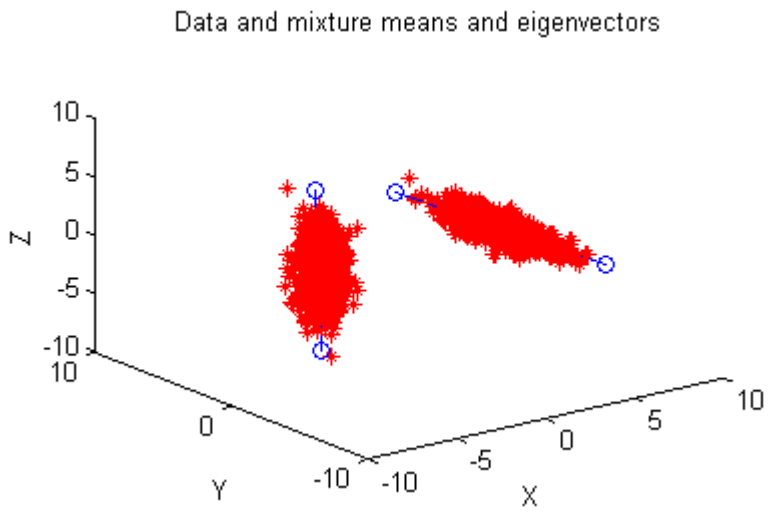


Figure 4. 3-D data with mixture means and eigenvectors

In the plot, the data vectors are plotted as stars and on the data we superimpose lines parallel to the two mixture component eigenvectors, and passing through their respective means. As can be seen, the two means of the mixture components converge to the center of the clusters, with the eigenvectors capturing the principal direction of the cluster.

The training procedure converges rapidly, in fact, convergence is obtained by the end of the third iteration. The squared error using the MPC is around 7~8 times smaller than with using the PCA, even with the same number of total parameters.

For this same random data we also train multiple eigenspaces using the VQPCA algorithm. In order to make a fair comparison we train two clusters with one eigenvector each, as for the MPC. The converged means and eigenvectors for the VQPCA are identical to those for the MPC, however the squared error using this representation is around twice as large as the MPC error. This improved error performance of the MPC is to be expected, as for the same set of means and eigenvectors, the VQPCA is a special case of the MPC, with the weights chosen so that one of the weights is one and the rest are all zero.

As mentioned before, we collected a face sequence with the person exhibiting three poses, looking left, looking straight ahead and looking left. We have shown the first three eigen-

coefficients of the face data from this sequence in Figure 2 to illustrate that the data belongs to multiple clusters. We now try to model this face data using a MPC with three components, each having two eigenvectors and compare this with using the PCA with six eigenvectors. When we examine the converged means and eigenvectors for the resulting mixture components we see that they do indeed correspond to the left pose, the center pose and the right pose. These parameters at convergence are shown in Figure 5.

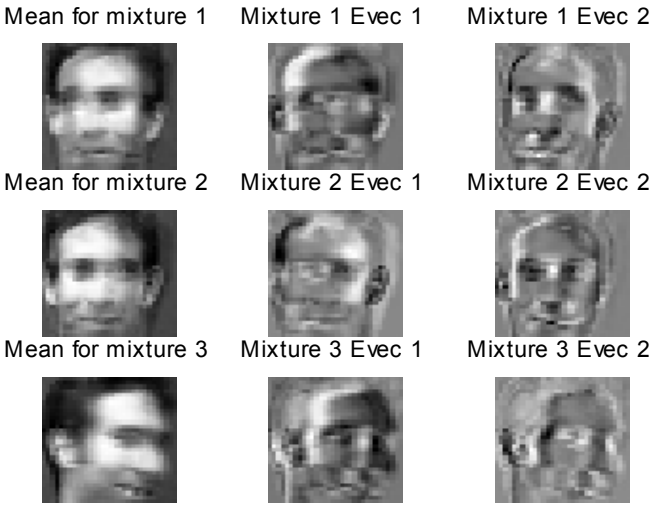


Figure 5. Means and eigenvectors for the MPC

We can see from Figure 5, that the means converge to the three poses, left, right and center. The eigenvectors highlight the dominant motion associated with each cluster. The parameters for the mixtures converge after around 5 iterations. The squared error for MPC with 3 component eigenspaces with two eigenvectors each is around 25% smaller than the error for the PCA with size eigenvectors. Hence the MPC has smaller error, even with the same number of total eigenvectors as the PCA.

4. Error Concealment using Mixtures of Principal Components

We have introduced an accurate and efficient linear model to capture object appearance variations. In order to examine the performance of the MPC for error concealment, we first need to simulate the errors created typically in wireless networks. We thus examine models to create bursty packet errors. The first model we consider simulates bursty packet loss using a two-state Markov chain, and was proposed by Yajnik et al [26]. This model is shown in Figure 6.

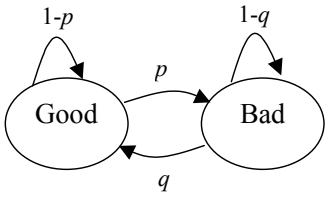


Figure 6. Two state Markov chain to simulate bursty packet loss

When the Markov chain is in the Bad state the received packet is dropped otherwise the packet is successfully decoded. The parameters p and q of the Markov chain are selected as described in [26], and affect the overall loss probability and the maximum burst size (MBS) of consecutive errors.

We also examine another model for bursty losses in wireless networks as proposed by Nguyen et al [27]. They propose a model for wireless transmission over the WaveLAN interface. Their model is also a two-state Markov chain, one corresponding to error-free conditions and the other corresponding to error. However in each state they generate a random number corresponding to the number of packets belonging to that state and at each clock tick they transition between states. For instance if they generate a number 20 in the good state, this means that the network sees 20 error free packets and if they generate 15 in the bad state, then the network sees 15 packets with error. We show the model in Figure 7.

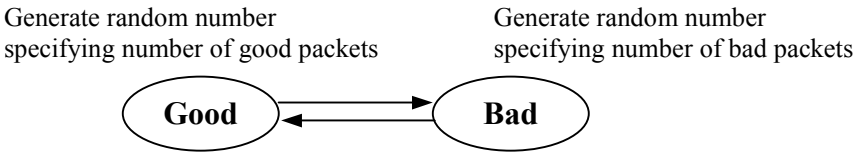


Figure 7. Model to simulate bursty packet losses over wireless networks

The focus of the work by Nguyen et al is on modeling the distribution of these random variables in each of the states accurately. We also use this loss model with the parameters derived by them while simulating down-link performance over a WaveLAN network.

For ease of implementation, in our experiments we use one packet to contain data for one 16×16 block. Even though the error is applied to such packets, it is possible to have consecutive blocks lost, so such a scheme can simulate the multiple blocks per packet case. It is important to realize that the scheme is not limited to such packet sizes, but may be applied across a variety of packet sizes.

In our error concealment experiments we first use the face sequence that we have collected. This is because the sequence consists of an object, the face, exhibiting multiple appearance variations due to pose changes. Also this object may be easily tracked, thereby

making it easy to build a model for it. Accurate tracking across frames with errors is required for such an error concealment scheme, as we need to segment out the object. However, performing some simple spatial interpolation before using the tracking algorithm can significantly improve tracking results. This is because most trackers use color and intensity based segmentation, so we do not need to recover the details in the missing blocks. We later also perform some experiments on the standard Foreman sequence, as it also contains an object, face again, showing appearance variations and use MPEG-4 segmentation information to track the face. Of course when such segmentation information is already available, we do not need to rely anymore on the tracking performance.

The first experiment we perform is on Intra coded frames. We perform model-based error concealment on the face sequence with 180 frames. We use both the MPC and the PCA as models for the face. We train the MPC and the PCA using 20 clean frames from the sequence and we use these trained eigenspaces to conceal errors in the remaining 160 frames. Such a learning based scheme requires representative training data to model the test data accurately. In practice, it is not always guaranteed that we will have access to clean and representative training data prior to transmission. Hence in such cases we need to train the model online, use received clean frames to update the model parameters. The decoder being aware of the presence or absence of errors makes it easy to identify clean frames. Another advantage of online training is the fact that the model can be adapted to the changing appearance of the object, for instance due to lighting or other changes. Some work on online training and updating of eigenspaces was done by Murakami and Kumar [28], and this is a future direction to improve such a model based scheme.

We perform two iterations of POCS to obtain better convergence results. As a measure of our error concealment performance we evaluate the PSNR between the frame with error concealment and the frame with no errors. In order to compare against currently used error concealment schemes we also perform error concealment by copying a missing block from the same location in the previous frame. We call this scheme Zero Motion error concealment.

We repeat the experiment across different loss probabilities and different quantization step sizes. We vary the loss probabilities in the range 0.05~0.3 and change the quantization step sizes in the range 1~16. We use varying quantization step sizes to illustrate the performance across different bit rates. Although a fixed quantization step size does not correspond to a fixed bit rate, we can use the quantization step size as indicative of the bit rate, with a large quantization step size corresponding to a low bit rate. In each case the PSNR is computed with respect to the quantized sequence, however with no errors. These results are shown in Figure 8.

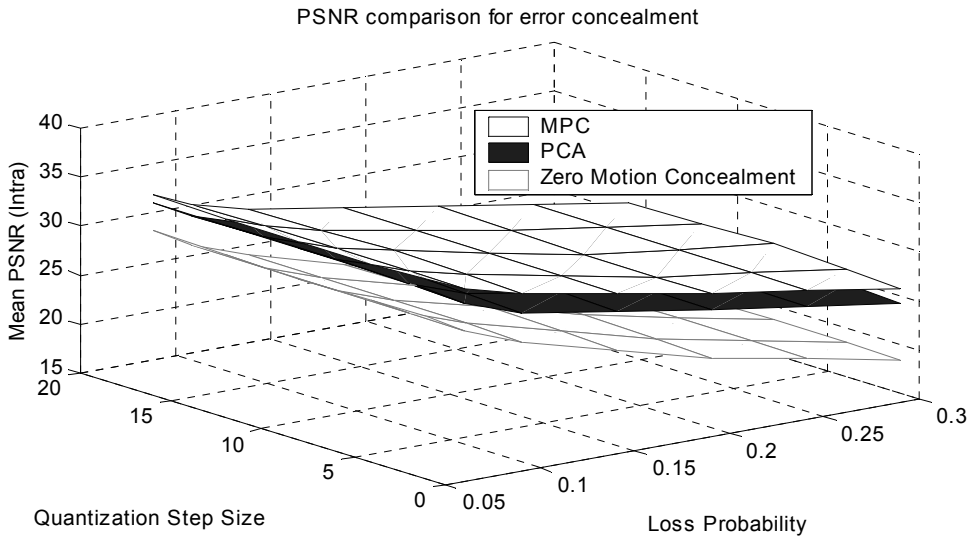


Figure 8. Error concealment across different loss probabilities and quantization step sizes (Intra)

We can see from Figure 8 that model based error concealment schemes consistently outperform the Zero Motion error concealment across all the different loss probabilities and quantization step sizes. In fact the improvement ranges between 4~9 dB with a larger improvement occurring at higher loss probabilities. Also, among the model-based concealment schemes, the MPC performs consistently better than the PCA by around 1~2 dB. This illustrates that the MPC is better able to capture the variations in the face appearance than the PCA. In terms of computational complexity, the concealment scheme takes increases the H.263 decoding time by around 9~12%, with more computations needed with larger error probabilities. This increase is only for frames with errors, for frames without errors, there is no computational overhead. Since H.263 decoders can typically operate at higher than the frame rate, this increase still allows for real-time decoding. We show sample frames with error concealment from this face sequence in Figure 9.

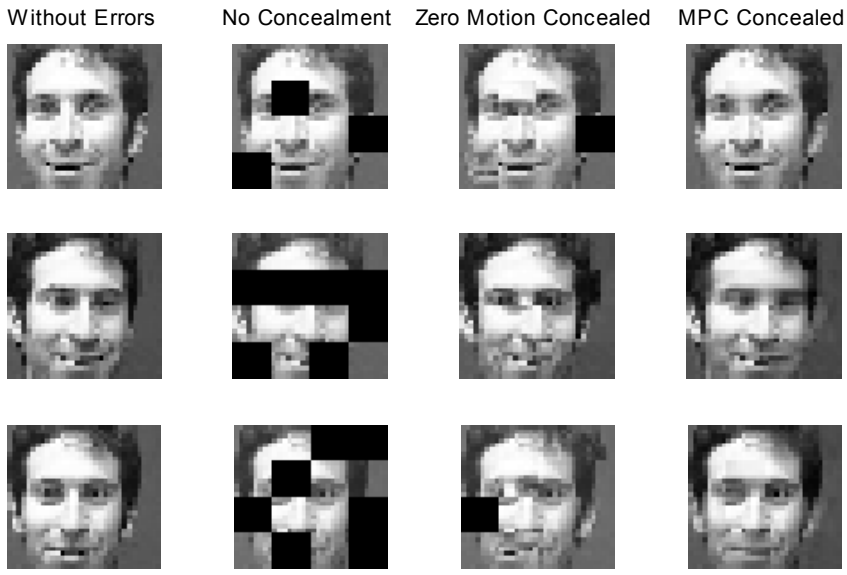


Figure 9. Error concealment using MPC for Intra case

In Figure 9 we show three sample frames with and without error concealment. The frames on the right have the errors concealed using the MPC as model. The no concealment case shows the missing blocks set to black, with the bursty nature of the errors indicated by the consecutive error blocks that occur on the faces. The Zero Motion concealed frames are also shown to illustrate the poor performance of such schemes. Some frames with Zero Motion concealment also contain dark blocks due to error propagation near the beginning of the sequence, as the first frame does not use any error concealment.

We then focus on error concealment for Inter coded sequences. For these sequences we assume that the motion vectors are available and packet loss corresponds to loss of the residue block. There are many advanced error resilient modes in both H.263 and MPEG-4 that allow for such availability of motion vectors under lossy conditions. Hence this means that we have the motion compensated prediction for the current frame. Using the motion compensated prediction is one of the error concealment schemes in practice currently. Although propagation of errors becomes a concern, still the visual artifacts are not as bad as when we lose data in the Intra case, especially when the sequence does not have too much arbitrary motion. Also, we allow for intra-refresh of blocks as recommended by the H.263 standard, and this helps to reduce the effect of error propagation. As opposed to this, the MPC and PCA based concealment schemes project the face into the respective eigenspaces and replace blocks in the face for which the residue block was lost. We perform two iterations of the POCS to obtain these results. In order to highlight the improvements of using a model based concealment scheme as opposed to using the motion

compensated prediction, we sub-sampled the sequence in time by a factor of three, thereby making the motion compensated prediction not as good as for the full frame-rate sequence.

As for the Intra case, we repeat the experiment across different loss rates and quantization step sizes. Since the original face sequence has smooth motion, the residues are typically small. Hence discarding the residue under lossy conditions does not degrade the video quality as much as in the Intra case. Hence a larger loss probability can be tolerated in this case, so we vary the loss probability in the 0.05~0.5. We vary the quantization step size again in the range 1~16. The results for error concealment are shown in Figure 10.

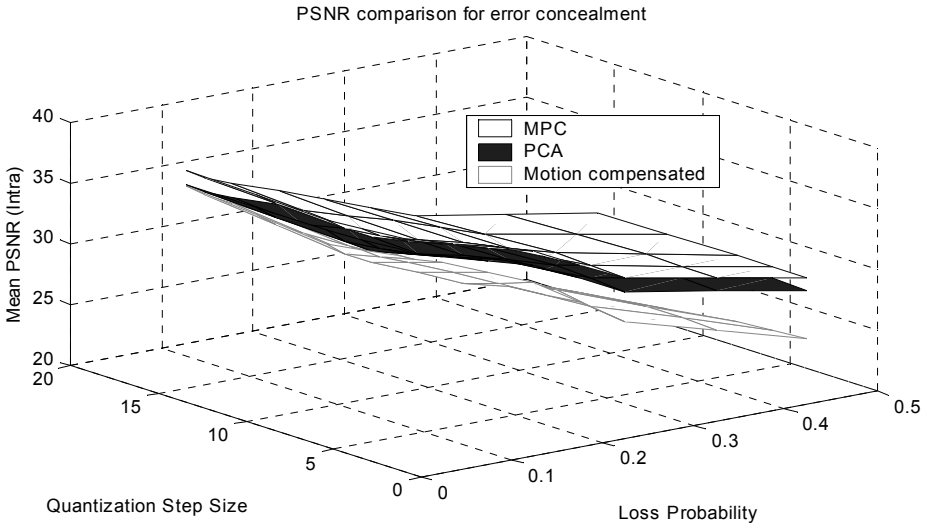
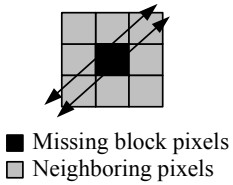


Figure 10. Error concealment across different loss probabilities and quantization step sizes (Inter)

We can see from Figure 10 that model-based error concealment outperforms the motion compensated error concealment by around 2~7 dB. Also, the improvements with the model-based schemes are more significant at higher loss probabilities thereby making these schemes more robust to large loss probabilities. Again, the MPC based error concealment outperforms the PCA based error concealment by around 1~2 dB. As for the Intra case, the concealment scheme increases the H.263 decoding time by around 11~13%, for frames with errors. We also perform intra-frame concealment using a simple version of the scheme proposed by Kwok and Sun [1]. We show this in Figure 11.



1. Use neighboring pixels to identify dominant edge directions (in 45° steps). Choose two dominant directions
2. Filter along those directions using weights inversely proportional to distance from missing pixel.
3. Combine filtered images

Figure 11. Intra frame concealment scheme

Clearly such an intra-frame scheme would work well when the missing block is spatially similar to the neighbors. However sometimes missing blocks contain detail information and features unavailable from the neighbors. Also, when we have bursty errors, the lack of good neighbors leads to poor performance of such schemes. We show an example of the error concealment performance using three consecutive frames in Figure 12. We choose these frames as we can see the effect of error propagation and intra refresh.

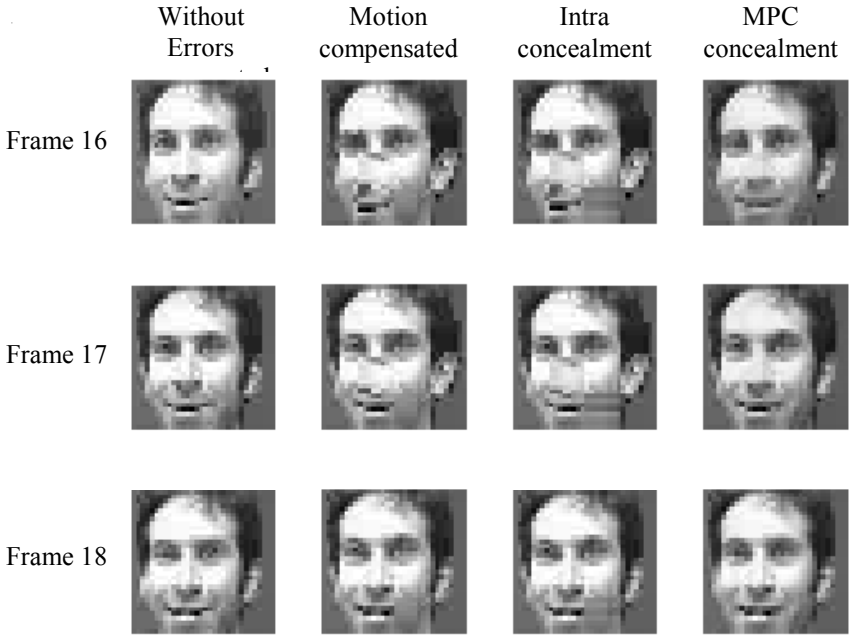


Figure 12. Sample frames with error concealment using MPC

We can see from Figure 12 that frames 16 and 17 not only have poor PSNR but the errors propagate between frame 16 and 17 due to the use of the motion compensated prediction, however due the presence of intra coded blocks and few errors in frame 18, the PSNR for the motion compensated concealment scheme improves. As against this, the frames with errors concealed using the MPC have a steadily high PSNR. As expected, this simple intra-frame concealment cannot recover the missing features. Also due to the dominant horizontal edge near the lips, the concealed blocks are of poor quality. On average this intra-frame concealment scheme performs 0.5~2 dB worse than the motion compensated concealment for our sequences.

We repeat all of the above experiments using the model proposed by Nguyen et al [27] and find that the results for error concealment are consistent with the two-state Markov chain proposed by Yajnik et al. We obtain an improvement of 4~8 dB using model based error concealment over the Zero Motion concealment for the Intra coded face sequence. Similarly, we

obtain an improvement over motion compensated error concealment by around 2~7 dB by using model based error concealment. Also, the MPC as a model consistently outperforms the PCA by around 1~2 dB across all these test scenarios.

We also performed some experiments with the Foreman sequence. We examine the first 160 frames of the sequence as these contain the person in the foreground. We use the MPEG-4 segmentation information to track the foreground. We crop out the head in an 80×80 box and use data from 20 frames to train our models, and use the rest for our error concealment experiments. We simulate bursty packet loss using the models described before. We find that over a loss probability range of 0.05~0.3 our model based error concealment schemes outperform motion compensation based schemes by around 2~5 dB. We show some sample frames from the sequence corresponding to a loss probability of 0.1, in Figure 13.

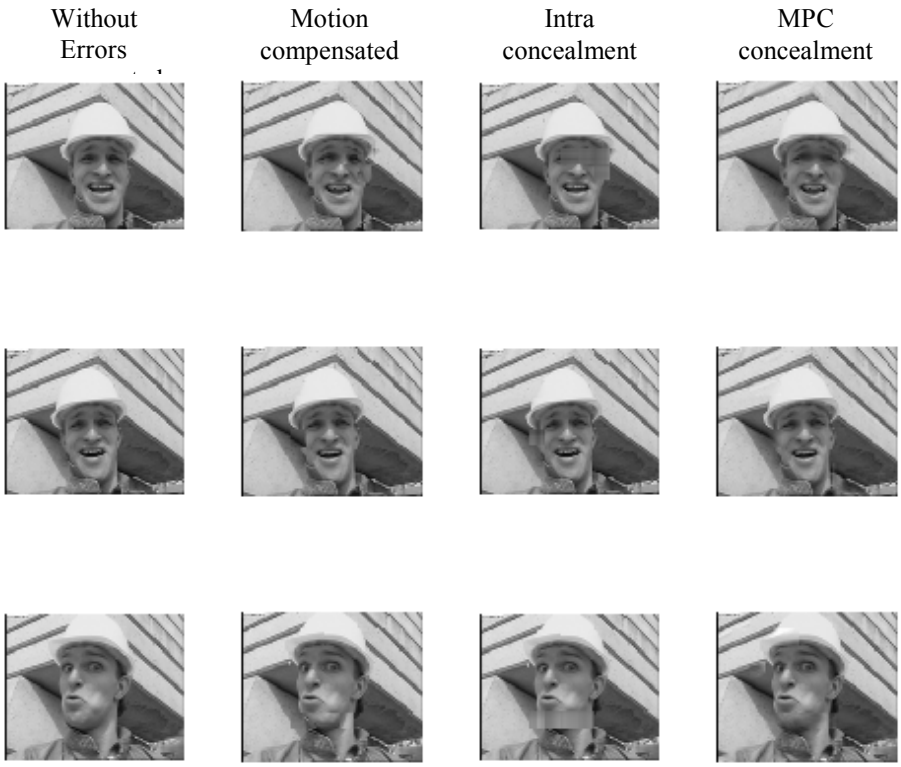


Figure 13. Sample frames from Foreman Sequence

Again, the simple intra-frame concealment does not perform well, as the missing blocks are not very similar to their neighbors, and performs 1~2 dB worse than motion compensated concealment. Due to a larger size of the region of interest for this sequence, the H.263 decoding time is increased by around 14~19% by this error concealment scheme.

5. Conclusion

We have introduced schemes for model-based error concealment. Such model-based error concealment schemes outperform previously proposed schemes since the models are trained specific to the object of interest and can thereby capture the statistical variations in the object appearance more effectively. Such model-based concealment approaches are very useful especially in conjunction with the MPEG-4 standard, which uses object based coding, since the video stream contains the segmented object, eliminating the need for tracking the object.

We focus on building accurate, efficient and computationally simple models for this task. We introduce a new statistical modeling technique called MPC that uses a mixture of eigenspaces to represent data. We show using simulated data that the MPC provides us better error performance than using the PCA, even when both use the same number of parameters. We then use these models for error concealment.

We perform concealment for both Intra coded as well as Inter coded face sequences across a variety of quantization step sizes and packet loss probabilities. We also examine two different models for simulating bursty losses over wireless networks. We show that across all these different test conditions and simulations, a model-based approach to error concealment outperforms conventional concealment techniques. We show that model-based error concealment leads to an improvement of 4~8 dB in PSNR over Zero Motion error concealment, and 2~7 dB over motion compensated error concealment. We show that the MPC based error concealment outperforms the PCA based error concealment by around 1~2 dB in PSNR, even with the same number of total parameters, as it captures the data variations more efficiently than the PCA. In terms of computational complexity, such a model based error concealment scheme increases the H.263 decoding by around 9~19% depending upon the size of the region of interest and the loss probability. This increase is only for frames with errors, for frames without errors, there is no computational overhead.

We realize that the MPC is a very general statistical framework for capturing data variations and so may be used for other tasks such as tracking, recognition, segmentation, etc. Some extension of the model needs to be done to allow for online update of the model parameters, which increases the flexibility, leading to better error concealment performance.

6. Acknowledgement

We are very grateful to the anonymous reviewers whose insightful comments have helped improve the paper.

7. Appendix

7.1. Solution for the means

We first rewrite the optimization equation from (5) as follows

$$\min_{\mathbf{m}_q} \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{x}_i - \hat{\mathbf{X}}_i \mathbf{w}_i \right\|^2 \tag{10}$$

We need to now expand $\hat{\mathbf{X}}_i$ in terms of the means before we can take derivatives and set to zero.

$$\hat{\mathbf{x}}_{ij} = \mathbf{U}_j \mathbf{U}_j^T \left(\mathbf{x}_i - \mathbf{m}_j \right) + \mathbf{m}_j = \left(\underbrace{\mathbf{I}_D - \mathbf{U}_j \mathbf{U}_j^T}_{\mathbf{A}_j} \right) \mathbf{m}_j + \underbrace{\mathbf{U}_j \mathbf{U}_j^T}_{\mathbf{B}_j} \mathbf{x}_i$$

$$\hat{\mathbf{X}}_i = \sum_{j=1}^{M} \hat{\mathbf{x}}_{ij} \mathbf{e}_j^T \text{ where } \mathbf{e}_j = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} j^{th} \text{position} \tag{11}$$

$$\text{Note : Since } \mathbf{U}_j^T \mathbf{U}_j = \mathbf{I}_M \Rightarrow \mathbf{A}_j^T \mathbf{A}_j = \mathbf{A}_j \text{ and } \mathbf{A}_j^T \mathbf{B}_j = \mathbf{0}$$

In the above equation $\mathbf{B}_j$ is the matrix that projects data onto the sub-space spanned by the eigenvectors in $\mathbf{U}_j$, while $\mathbf{A}_j$ is the matrix that projects data onto the space orthogonal to this sub-space. We may use equation (11) to replace $\hat{\mathbf{X}}_i$ in equation (10) and hence, the optimization criterion may be rewritten as follows.

$$\min_{\mathbf{m}_q} \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{x}_i - \left[\sum_{j=1}^{M} \left(\mathbf{A}_j \mathbf{m}_j + \mathbf{B}_j \mathbf{x}_i \right) \mathbf{e}_j^T \right] \mathbf{w}_i \right\|^2 \tag{12}$$

$$\text{where } \|\bullet\|^2 = (\bullet)^T (\bullet)$$

We may now expand equation (12) into individual terms and rewrite it as follows.

$$\min_{\mathbf{m}_q} \frac{1}{N} \sum_{i=1}^{N} \left[\begin{array}{l} \mathbf{x}_i^T \mathbf{x}_i - 2\mathbf{w}_i^T \left[\sum_{j=1}^{M} \left(\mathbf{A}_j \mathbf{m}_j + \mathbf{B}_j \mathbf{x}_i \right) \mathbf{e}_j^T \right]^T \mathbf{x}_i + \\ \mathbf{w}_i^T \left[\sum_{j=1}^{M} \left(\mathbf{A}_j \mathbf{m}_j + \mathbf{B}_j \mathbf{x}_i \right) \mathbf{e}_j^T \right]^T \left[\sum_{k=1}^{M} \left(\mathbf{A}_k \mathbf{m}_k + \mathbf{B}_k \mathbf{x}_i \right) \mathbf{e}_k^T \right] \mathbf{w}_i \end{array} \right] \tag{13}$$

We now drop the terms independent of $\mathbf{m}_j$ and expand the inner product terms.

$$\min_{\mathbf{m}_q} \frac{1}{N} \sum_{i=1}^{N} \begin{bmatrix} -2\sum_{j=1}^{M} \mathbf{w}_i^T \mathbf{e}_j \mathbf{m}_j^T \mathbf{A}_j^T \mathbf{x}_i + \sum_{j=1}^{M}\sum_{k=1}^{M} \mathbf{w}_i^T \mathbf{e}_j \mathbf{m}_j^T \mathbf{A}_j^T \mathbf{A}_k \mathbf{m}_k \mathbf{e}_k^T \mathbf{w}_i \\ \sum_{j=1}^{M}\sum_{k=1}^{M} \mathbf{w}_i^T \mathbf{e}_j \mathbf{m}_j^T \mathbf{A}_j^T \mathbf{B}_k \mathbf{x}_i \mathbf{e}_k^T \mathbf{w}_i + \sum_{j=1}^{M}\sum_{k=1}^{M} \mathbf{w}_i^T \mathbf{e}_j \mathbf{x}_i^T \mathbf{B}_j^T \mathbf{A}_k \mathbf{m}_k \mathbf{e}_k^T \mathbf{w}_i \end{bmatrix} \tag{14}$$

Notice that $\mathbf{w}_i^T \mathbf{e}_j = \mathbf{e}_j^T \mathbf{w}_i = \mathbf{w}_{ij}$, which are scalars and so may be moved to the front of each term. We now take derivatives with respect to $\mathbf{m}_q$ and set the result to zero.

$$\sum_{i=1}^{N} \begin{bmatrix} -2w_{iq} \mathbf{A}_q^T \mathbf{x}_i + 2\sum_{j=1, j\neq q}^{M} w_{iq} w_{ij} \mathbf{A}_q^T \mathbf{A}_j \mathbf{m}_j + \\ 2w_{iq}^2 \mathbf{A}_q^T \mathbf{A}_q \mathbf{m}_q + 2\sum_{k=1}^{M} w_{iq} w_{ik} \mathbf{A}_q^T \mathbf{B}_k \mathbf{x}_i \end{bmatrix} = \mathbf{0} \tag{15}$$

Now we may use the properties that $\mathbf{U}_j^T \mathbf{U}_j = \mathbf{I}_M \Rightarrow \mathbf{A}_j^T \mathbf{A}_j = \mathbf{A}_j \;\&\; \mathbf{A}_j^T \mathbf{B}_j = \mathbf{0}$ to simplify equation (15) as follows.

$$\mathbf{A}_q^T \mathbf{m}_q = \frac{1}{\sum_{i=1}^{N} w_{iq}^2} \mathbf{A}_q^T \left[\sum_{i=1}^{N} w_{iq} \left(\mathbf{x}_i - \sum_{j=1, j\neq q}^{M} w_{ij} \hat{\mathbf{x}}_{ij} \right) \right] \tag{16}$$

Equation (16) has multiple solutions due to the fact that $\mathbf{A}_j$ is singular. One solution for the means is as follows.

$$\mathbf{m}_q = \frac{1}{\sum_{i=1}^{N} w_{iq}^2} \left[\sum_{i=1}^{N} w_{iq} \left(\mathbf{x}_i - \sum_{j=1, j\neq q}^{M} w_{ij} \hat{\mathbf{x}}_{ij} \right) \right] \tag{17}$$

Clearly, any vector orthogonal to $\mathbf{A}_q$ may be added to $\mathbf{m}_q$ as defined in equation (17) to still remain a valid solution to equation (16). Thus, the means are allowed to move within the sub-space (hyper-plane) spanned by the eigenvectors in $\mathbf{U}_q$. This non-uniqueness does not affect the reconstruction error, as it is measured as a distance from the projection onto the hyper-plane, which is independent of the location of the mean within this hyper-plane. As an illustration we show a simple example scenario in Figure 14.

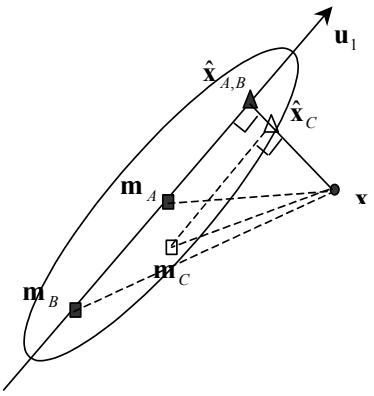


Figure 14. Projection onto eigenspace as function of mean location

In the example scenario we consider a single eigenspace with one principal direction $\mathbf{u}_1$, shown in the figure as the principal axis of the ellipse. When we project data point $\mathbf{x}$ onto this eigenspace, whether the mean is located at position $\mathbf{m}_A$ or at position $\mathbf{m}_B$, the resulting projection is $\hat{\mathbf{x}}_{A,B}$. In fact this is always the resulting projection as long as the location of the mean changes along the line specified by $\mathbf{u}_1$ passing through $\mathbf{m}_A$. As an illustration when the mean moves to location $\mathbf{m}_C$, the resulting projection also changes to $\hat{\mathbf{x}}_C$, which is different from $\hat{\mathbf{x}}_{A,B}$, thereby leading to a different reconstruction error. Hence we can tolerate this non-uniqueness of the solution for the mean, as long as it is limited to within the hyper-plane spanned by the eigenvectors.

7.2. Solution for Eigenvectors

We may now use the updated means while deriving the eigenvectors. The optimization criterion from equation (5) may be rewritten below in equation (18).

$$\min_{\mathbf{u}_{jk}} \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{x}_i - \sum_{j=1}^{M} w_{ij} \left[\mathbf{m}_j + \sum_{k=1}^{P} \left[\left(\mathbf{x}_i - \mathbf{m}_j \right)^T \mathbf{u}_{jk} \right] \mathbf{u}_{jk} \right] \right\|^2 \tag{18}$$

Let us define $c_{ijk} = \left(\mathbf{x}_i - \mathbf{m}_j \right)^T \mathbf{u}_{jk}$. Using this, the minimization criterion of (18) may be written as follows.

$$\min_{\mathbf{u}_{rs}} \frac{1}{N} \sum_{i=1}^{N} \left[\begin{array}{l} \mathbf{x}_i^T \mathbf{x}_i - 2\mathbf{x}_i^T \sum_{j=1}^{M} w_{ij} \left(\mathbf{m}_j + \sum_{k=1}^{P} c_{ijk} \mathbf{u}_{jk} \right) + \\ \sum_{j=1}^{M} \sum_{a=1}^{M} w_{ij} w_{ia} \left(\mathbf{m}_j + \sum_{k=1}^{P} c_{ijk} \mathbf{u}_{jk} \right)^T \left(\mathbf{m}_a + \sum_{b=1}^{P} c_{iab} \mathbf{u}_{ab} \right) \end{array} \right] \tag{19}$$

We now expand the terms in equation (19) and get the following.

$$\min_{\mathbf{u}_{rs}} \frac{1}{N}\sum_{i=1}^{N}\left[\begin{array}{l}\mathbf{x}_i^T\mathbf{x}_i - 2\mathbf{x}_i^T\sum_{j=1}^{M} w_{ij}\left(\mathbf{m}_j + \sum_{k=1}^{P} c_{ijk}\mathbf{u}_{jk}\right) + \sum_{j=1}^{M}\sum_{a=1}^{M} w_{ij}w_{ia}\mathbf{m}_j^T\mathbf{m}_a + \\ 2\sum_{j=1}^{M}\sum_{a=1}^{M} w_{ij}w_{ia}\mathbf{m}_j^T\sum_{b=1}^{P} c_{iab}\mathbf{u}_{ab} + \sum_{j=1}^{M}\sum_{a=1}^{M} w_{ij}w_{ia}\sum_{k=1}^{P}\sum_{b=1}^{P} c_{ijk}c_{iab}\mathbf{u}_{jk}^T\mathbf{u}_{ab}\end{array}\right] \tag{20}$$

Now we may drop all terms independent of $\mathbf{u}_{jk}$ and rewrite equation (20) using the knowledge that all eigenvectors of one mixture component are orthogonal to each other. Note, however that eigenvectors from different mixture components are not constrained to be orthogonal.

$$\min_{\mathbf{u}_{rs}} \frac{1}{N}\sum_{i=1}^{N}\left[\begin{array}{l}-2\sum_{j=1}^{M} w_{ij}\sum_{k=1}^{P}\mathbf{u}_{jk}^T\left(\mathbf{x}_i - \mathbf{m}_j\right)\mathbf{x}_i^T\mathbf{u}_{jk} + 2\sum_{j=1}^{M}\sum_{a=1}^{M} w_{ij}w_{ia}\sum_{b=1}^{P}\mathbf{u}_{ab}^T\left(\mathbf{x}_i - \mathbf{m}_a\right)\mathbf{m}_j^T\mathbf{u}_{ab} + \\ \sum_{j=1}^{M}\sum_{a=1,a\neq j}^{M} w_{ij}w_{ia}\sum_{k=1}^{P}\sum_{b=1}^{P} c_{ijk}\mathbf{u}_{ab}^T\left(\mathbf{x}_i - \mathbf{m}_a\right)\mathbf{u}_{jk}^T\mathbf{u}_{ab} + \\ \sum_{j=1}^{M} w_{ij}^2\sum_{k=1}^{P}\mathbf{u}_{jk}^T\left(\mathbf{x}_i - \mathbf{m}_j\right)\left(\mathbf{x}_i - \mathbf{m}_j\right)^T\mathbf{u}_{jk}\end{array}\right] \tag{21}$$

We also need to add the constraint that the eigenvectors need to have a norm of 1, and we may use the Lagrange multiplier method to add in the constraint to obtain equation (22).

$$\min_{\mathbf{u}_{rs}} \frac{1}{N}\sum_{i=1}^{N}\left[\begin{array}{l}-2\sum_{j=1}^{M} w_{ij}\sum_{k=1}^{P}\mathbf{u}_{jk}^T\left(\mathbf{x}_i - \mathbf{m}_j\right)\mathbf{x}_i^T\mathbf{u}_{jk} + \\ 2\sum_{j=1}^{M}\sum_{a=1}^{M} w_{ij}w_{ia}\sum_{b=1}^{P}\mathbf{u}_{ab}^T\left(\mathbf{x}_i - \mathbf{m}_a\right)\mathbf{m}_j^T\mathbf{u}_{ab} + \\ \sum_{j=1}^{M}\sum_{a=1,a\neq j}^{M} w_{ij}w_{ia}\sum_{k=1}^{P}\sum_{b=1}^{P} c_{ijk}\mathbf{u}_{ab}^T\left(\mathbf{x}_i - \mathbf{m}_a\right)\mathbf{u}_{jk}^T\mathbf{u}_{ab} + \\ \sum_{j=1}^{M} w_{ij}^2\sum_{k=1}^{P}\mathbf{u}_{jk}^T\left(\mathbf{x}_i - \mathbf{m}_j\right)\left(\mathbf{x}_i - \mathbf{m}_j\right)^T\mathbf{u}_{jk}\end{array}\right] + \lambda\left(\mathbf{u}_{rs}^T\mathbf{u}_{rs} - 1\right) \tag{22}$$

We may now take derivative with respect to $\mathbf{u}_{rs}$ and set the result to zero to obtain equation (23). Note that the term in the third line of equation (22) has two terms containing $\mathbf{u}_{rs}$, one when $a = r$ and $b = s$ and the other when $j = r$ and $k = s$, which do not happen together due to the $a \neq j$ condition. Both these terms are identical and their derivative may be grouped together as shown below in equation (23).

$$\frac{1}{N}\sum_{i=1}^{N}\begin{bmatrix} -2w_{ir}\left[(\mathbf{x}_i-\mathbf{m}_r)\mathbf{x}_i^T+\mathbf{x}_i(\mathbf{x}_i-\mathbf{m}_r)^T\right]\mathbf{u}_{rs}+ \\ 2\sum_{j=1}^{M}w_{ij}w_{ir}\left[(\mathbf{x}_i-\mathbf{m}_r)\mathbf{m}_j^T+\mathbf{m}_j(\mathbf{x}_i-\mathbf{m}_r)^T\right]\mathbf{u}_{rs}+ \\ 2\sum_{j=1,j\neq r}^{M}w_{ij}w_{ir}\sum_{k=1}^{P}\left[\mathbf{u}_{jk}^T(\mathbf{x}_i-\mathbf{m}_j)\right]\left[(\mathbf{x}_i-\mathbf{m}_r)\mathbf{u}_{jk}^T+\mathbf{u}_{jk}(\mathbf{x}_i-\mathbf{m}_r)^T\right]\mathbf{u}_{rs}+ \\ 2w_{ir}^2(\mathbf{x}_i-\mathbf{m}_r)(\mathbf{x}_i-\mathbf{m}_r)^T\mathbf{u}_{rs} \end{bmatrix}=-2\lambda\mathbf{u}_{rs} \quad (23)$$

This is clearly an eigenvector, eigenvalue problem that may be written as $\mathbf{C}_r\mathbf{u}_{rs}=\lambda\mathbf{u}_{rs}$ where the matrix $\mathbf{C}_r$ is defined as follows.

$$\mathbf{C}_r=\frac{1}{N}\sum_{i=1}^{N}\begin{bmatrix} w_{ir}\left[(\mathbf{x}_i-\mathbf{m}_r)\mathbf{x}_i^T+\mathbf{x}_i(\mathbf{x}_i-\mathbf{m}_r)^T\right]- \\ \sum_{j=1}^{M}w_{ij}w_{ir}\left[(\mathbf{x}_i-\mathbf{m}_r)\mathbf{m}_j^T+\mathbf{m}_j(\mathbf{x}_i-\mathbf{m}_r)^T\right]- \\ \sum_{j=1,j\neq r}^{M}w_{ij}w_{ir}\sum_{k=1}^{P}\left[\mathbf{u}_{jk}^T(\mathbf{x}_i-\mathbf{m}_j)\right]\left[(\mathbf{x}_i-\mathbf{m}_r)\mathbf{u}_{jk}^T+\mathbf{u}_{jk}(\mathbf{x}_i-\mathbf{m}_r)^T\right]- \\ w_{ir}^2(\mathbf{x}_i-\mathbf{m}_r)(\mathbf{x}_i-\mathbf{m}_r)^T \end{bmatrix} \quad (24)$$

The first P eigenvectors of this matrix are the desired eigenvectors of the mixture component.

References

[1] W. Kwok and H. Sun, "Multi-directional interpolation for spatial error concealment," *IEEE Trans. Consumer Electronics.*, vol. 39, no. 3, p. 455-60, August 1993.

[2] S. Aign and K. Fazel, "Temporal and spatial error concealment techniques for hierarchical MPEG-2 video codec," Proc. Globecom, p. 1778-83, 1995.

[3] H. Sun and W. Kwok, "Concealment of damaged block transform coded images using projections onto convex sets," *IEEE Trans. Image Proc.*, vol. 4, no. 4, p. 470-77, April 1995.

[4] W-M. Lam and Amy Reibman, "An error concealment algorithm for images subject to channel errors," *IEEE Trans. Image Processing*, vol. 4, no. 5, May 1995.

[5] M-J. Chen, L-G. Chen and R-M. Weng, "Error concealment of lost motion vectors with overlapped motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol.7, no. 3, p. 560-63, June 1997.

[6] J. W. Park, J. W. Kim and S. U. Lee, "DCT coefficients recovery-based error concealment technique and its application to the MPEG-2 bit stream error," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 6, Dec. 1997.

[7] L. Atzori, F. G. B. De Natale, "Error concealment in video transmission over packet networks by a sketch based approach," *Signal Processing: Image Communication*, vol. 15, pp. 57-76, 1999.

[8] S. Tsekeridou and I. Pitas, "MPEG-2 error concealment based on block-matching principles," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 4, June 2000.

[9] S. Shirani, B. Erol and F. Kossentini, "A concealment method for shape information in MPEG-4 coded sequences," *IEEE Trans. Multimedia*, vol. 2, no. 3, Sept. 2000.

[10] C.-H. Chou and C.-W. Chen, "A perceptually optimized 3-D subband codec for video communication over wireless channels," *IEEE Trans. Circuits Syst. Video Technol.*, vol.6, no. 2, April 1996.

[11] H. Man, F. Kossentini and M. T. J. Smith, "A family of efficient and channel error resilient wavelet/subband image coders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, Feb. 1999.

[12] J. Lu, K. B. Letaief, M. L. Liou, "Robust video transmission over correlated mobile fading channels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 5, Aug. 1999.

[13] J. Zhang, J. F. Arnold and M. R. Frater, " A cell-loss concealment technique for MPEG-2 coded video," *IEEE Trans. Circuits Syst. Video Technol*., vol. 10, no. 4, June 2000.

[14] K. Aizawa and T. S. Huang, "Model-based image coding: Advanced video coding techniques for very low bit-rate applications," *Proc. IEE*E, vol. 83, pp. 259–271, Feb. 1995.

[15] D. E. Pearson, "Developments in model-based video coding," *Proc. IEE*E, vol. 83, pp. 892–906, June 1995.

[16] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. Wiley and Sons, New York, NY, 1973.

[17] T. Hastie and W. Stuetzle, "Principal curves," *Journal of the American Statistical Association*, vol. 84, pp. 502-16, 1989.

[18] E. Oja, "Neural networks, principal components and subspaces," *International Journal of Neural Systems*, vol. 1, pp. 61-68, 1989.

[19] S. Y. Kung and K. I. Diamantaras, "A neural network learning algorithm for adaptive principal component extraction (APEX)," Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 861-64, 1990.

[20] T. Cox and M. Cox, *Multidimensional Scaling*, Chapman and Hall, London 1994.

[21] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323-26, December 2000.

[22] N. Kambhatla and T. K. Leen, “Dimension reduction by local principal component analysis,” *Neural Computation*, vol. 9, pp. 1493-1516, 1997.

[23] M. Tipping and C. M. Bishop, “Mixtures of probabilistic principal component analyzers,” *Neural Computation*, vol. 11, no. 2, pp. 443-82, 1999.

[24] M. I. Sezan and M. Tekalp, “Adaptive image restoration with artifact suppression using the theory of convex projections,” *IEEE Trans. Acoust. Speech and Signal Processing*, vol. 38, no. 1, pp. 181-5, January 1990.

[25] J. Huang and T. Chen, "Tracking of Multiple Faces for Human-Computer Interfaces and Virtual Environments", IEEE Intl. Conf. on Multimedia and Exposition, New York, August 2000.

[26] M. Yajnik, S. Moon, J. Kurose, D. Towsley, “Measurement and modeling of the temporal dependence in packet loss,” Proc. IEEE INFOCOM, pp. 345-52, March 1999.

[27] G. T. Nguyen, R. H. Katz, B. Noble and M. Satyanarayanan, “A trace-based approach for modeling wireless channel behavior,” Proc. 1996 Winter Simulation Conference, Coronado, CA, Dec. 1996.

[28] H. Murakami, B.V.K.V. Kumar, “Efficient calculation of primary images from a set of images,” *IEEE Trans. Pattern Anal. Machine Intelligence*, vol. 4, no. 5, pp. 511-15, September 1982.