WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. 2002; **2**:607–624 (DOI: 10.1002/wcm.83)

Second-generation error concealment for video transport over error-prone channels

Trista Pei-chun Chen and Tsuhan Chen*,†

Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh
PA
U.S.A.

Summary

Video transport over error-prone channels may result in loss or erroneous decoding of the video data. Error concealment is an effective mechanism to reconstruct the video data. In this paper, we review error-concealment methods and introduce a new framework, which we refer to as *second-generation error concealment*. All the error-concealment methods reconstruct the lost video data by making use of certain *a priori* knowledge about the video content. *First-generation error concealment* builds such *a priori* in a heuristic manner. The proposed second-generation error concealment builds the *a priori* by modeling the statistics of the video content explicitly, typically in the region of interest (ROI). Context-based models are trained with the correctly received video data and then used to replenish the lost video data. Trained models capture the statistics of the video content and thus reconstruct the lost video data better than reconstruction by heuristics. A new dynamic model 'updating principal components' (UPC) is proposed as a model for second-generation error concealment. UPC can be applied to pixel values to conceal loss of pixel data. In addition, UPC can be applied to motion vectors, which results in 'updating eigenflows' (U-Eigenflow), to conceal loss of motion vectors. With UPC applied to both pixel values and motion vectors, hybrid temporal/spatial error concealment can be achieved. The proposed second-generation error-concealment method provides superior performances to first-generation error-concealment methods. Copyright © 2002 John Wiley & Sons, Ltd.

*Correspondence to: Tsuhan Chen, Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, U.S.A.
†E-mail: tsuhan@cmu.edu

Contract/grant sponsor: Industrial Technology Research Institute

KEY WORDS

second-generation error concealment
first-generation error concealment
updating principal components (UPC)
updating eigenflows (U-Eigenflow)
Projection onto Convex Sets (POCS)
error concealment
error resilient video coding
wireless multimedia

1. Introduction

When transmitting video data over error-prone channels, the video data may suffer from losses or errors. Error concealment is an effective way to recover the lost information at the decoder. Compared to other error-control mechanisms such as forward error correction (FEC) [1] and automatic retransmission request (ARQ) [2], error concealment has the advantages of not consuming extra bandwidth as FEC and not introducing retransmission delay as ARQ. On the other hand, error concealment can be used to supplement FEC and ARQ when both FEC and ARQ fail to overcome the transmission errors [3].

Error concealment is performed after error detection. That is, error concealment needs to be preceded with some error-detection mechanism to know where the errors in the decoded video are located. For example, error detection provides information as which part of the received video bitstream is corrupted. Various methods, such as checking the video bitstream syntax, monitoring the packet numbers of the received video data, and so on, can be applied [4,5]. In this paper, we assume that the errors are located and that such information is available to us. We focus on the reconstruction for the lost video.

In general, spatial, spectral, or temporal redundancies of the received video data are utilized to perform error concealment [6]. Hybrid or dynamic switching of spatial/temporal error-concealment methods is also possible [7–9]. In this paper, we will review these error-concealment methods.

All error-concealment methods reconstruct the lost video content by making use of some *a priori* knowledge about the video content. Most existing error-concealment methods, which we refer to as *first-generation error concealment*, build such *a priori* in a heuristic manner by assuming smoothness or continuity of the pixel values, and so on. The proposed *second-generation error concealment* methods train context-based models as the *a priori*. Methods of such a framework have advantages over first-generation error concealment, as the context-based model is created specifically for the video content and hence can capture the statistical variations of the content more effectively.

It is important for a second-generation error-concealment approach to choose a model that can represent the video content effectively. Principal component analysis (PCA) has long been used to model visual content of images. The most well-known example is using eigenfaces to represent human faces [10]. In this paper, we introduce a new dynamic model 'updating principal components' (UPC) [11] for second-generation error concealment. UPC is very suitable for error-concealment applications in that it updates with nonstationary video data. UPC can be applied to pixel values in regions of interest (ROI) in video frames. In addition, UPC can be applied to motion vectors (MVs), which results in '*updating eigenflows*' (*U-Eigenflow*). With both UPC for pixel values and UPC for MVs, hybrid temporal/spatial error concealment can be achieved.

This paper is organized as follows. In Section 2, we review *first-generation error concealment* by providing a survey of conventional error-concealment methods. We introduce the new framework of *second-generation error concealment* in Section 3. A detailed description of using UPC for error concealment is provided in Section 4. Both UPC for pixel values and for MVs will be discussed. We conclude in Section 5.

2. First-generation Error Concealment

Error concealment relies on some *a priori* to reconstruct the lost video content. First-generation error-concealment methods build the *a priori* for reconstructing the lost video content in a heuristic manner. A simple example is to assume the pixel values to be smooth across the boundary of the lost and retained regions. Methods of this framework assume smoothness or continuity of the video data in different domains such as spatial, spectral, temporal, or some transforms of these domains. To recover lost data with the smoothness assumption, interpolation or optimization based on certain objective functions are often used. Since first-generation error-concealment methods perform error concealment with such heuristic knowledge, we also call them *heuristic-based error concealment*.

According to the domains in which smoothness assumptions are applied, first-generation error-concealment methods fall into two categories: spatial/spectral and temporal, as follows.

2.1. Spatial/spectral Error Concealment

Spatial error concealment assumes that images are smooth in nature. Lost image content can be reconstructed by interpolation of the neighboring pixels. Work by Wang *et al.* [12] and Hemami and Meng [13] are earlier examples of using spatial interpolation to accomplish the task of error concealment. However, spatial interpolation approaches often suffer from blurring in the edges of the image. Several approaches have been proposed to solve this problem. Suh and Ho [14] proposed to find edges first and interpolate along the edge directions. Zhu *et al.* [15] proposed to use a second-order derivative-based method to reduce the blur across the edge while enforcing the smoothness along the edge. Zeng and Liu [16] proposed to perform directional interpolation based on the neighbor's geometric structure. Robie and Mersereau [17] proposed to use the Hough transform to determine the best orientation for either directional filtering or interpolation. Interpolation can be applied not only to the spatial domain but also to the spectral domain such as the discrete cosine transform (DCT) domain, as proposed by Chung *et al.* [18]. Some other methods are based on projection onto convex sets (POCS), which iteratively uses the smoothness assumption and pixel value or DCT coefficient range information for error concealment [19,20].

An extension to the assumption that natural images are smooth and the values are continuous spatially or spectrally is to assume that images can be modeled by Markov random fields (MRF) [21]. MRF-based error-concealment methods were first proposed by Salama *et al.* [22–24]. Later Shirani *et al.* [25] proposed to adaptively adjust the MRF model parameters without increasing the model order and showed that the adaptive MRF outperformed conventional MRF methods. Multiscale MRF (MMRF) by Zhang and Ma [26] is another extension of MRF. MMRF models image blocks instead of image pixels. Work by Zhang *et al.* [27] models the DCT coefficients as a first-order Markov process and uses Laplacian distribution to model the density function of the DCT coefficients.

2.2. Temporal Error Concealment

Temporal error-concealment methods use the temporal neighbor, that is, the previous frame or the next frame, to conceal the loss of the current frame. Temporal error-concealment methods assume the video content to be smooth or continuous in time. A basic approach is to replace the lost block of the current frame with the content of the previous frame at the same block location. An advanced approach is to replace the lost block with the content of the previous frame at the motion-compensated location. This advanced temporal error-concealment scheme needs motion vector information to find the corresponding block location in the previous frame. However, in the process of transmission, MVs can be lost as well. Without MVs, temporal error concealment with motion compensation cannot provide satisfactory reconstruction results. Therefore, techniques to estimate the lost MVs are explored. Boundary matching algorithm (BMA) proposed by Lam *et al.* [28] is a popular method to estimate lost MVs. Extensions to BMA can be found in References [29 to 32]. Decoder motion vector estimation (DMVE) proposed by Zhang *et al.* [33,34] treats the loss of MVs as a motion estimation problem, in the decoder instead of in the encoder. Motion field interpolation (MFI) and its extensions proposed by Al-Mualla *et al.* [35,36] estimate the MVs from neighbors with a single or multiple reference frames. Furthermore, Lee *et al.* [37] extended translational block motion to affine transform for motion-compensated error concealment.

3. Second-generation Error Concealment

Second-generation error concealment goes beyond heuristics. A more sophisticated way to obtain the *a priori* for error concealment is through training. Second-generation error concealment builds the *a priori* by training a context-based model for an object or ROI and uses this trained model to recover the lost data. With object-based video coding standards such as MPEG-4 [38], the video bitstream already contains object, or ROI, information, which makes second-generation error concealment possible. In cases in which the ROI information is not available in the video bitstream, object trackers [39] can be used to extract the ROI information. Figure 1 shows three video frames with face regions tracked and specified as the ROI. Since the context-based model is created specifically for the object, it captures the statistical variations of the object effectively and yields good concealment results. Since second-generation error concealment methods train and apply context-based models for error concealment, we also call them *model-based error concealment*.

As mentioned, PCA has been widely used to model image statistics. Face images, the most common examples of ROI, especially in video telephony or videoconferencing, can be modeled well with PCA. Figure 2 shows an example of using PCA with a mean and two eigenvectors to represent face images. The mean captures the average face appearance and the eigenvectors characterize variations such as pose or expression variations.

With PCA as the model to describe the object statistics, we can train the PCA model, that is, the mean and the eigenvectors, with pixel values in the ROI from correctly received frames. Then, we project any corrupted ROI to the PCA model to recover the lost data in the corrupted ROI. Using face images as an example of ROI, we illustrate such an error-concealment scheme in Figure 3. The PCA model shown in ② is trained in advance. The corrupted ROI shown in ① is projected to the PCA model to get the recovered ROI as shown in ③.

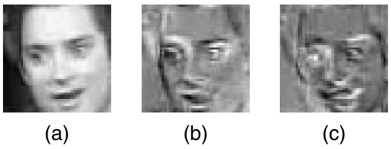

Fig. 2. PCA for face images: (a) mean; (b) first eigenvector; and (c) second eigenvector.

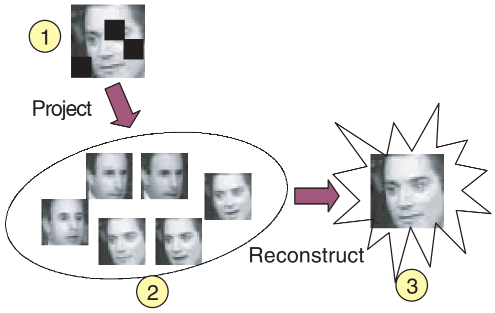

Fig. 3. Error concealment with PCA.

POCS can be used in the second-generation error-concealment scheme based on PCA modeling of the ROI. Error concealment based on POCS formulates each constraint about the unknowns as a convex set. The optimal solution is obtained by iteratively projecting the previous solution onto each convex set. The projections refer to (i) projecting the data with some losses to the PCA model that is built on error-free data, and (ii) replacing the projection result with the correctly received data in the corresponding region. Illustration of POCS-based error concealment

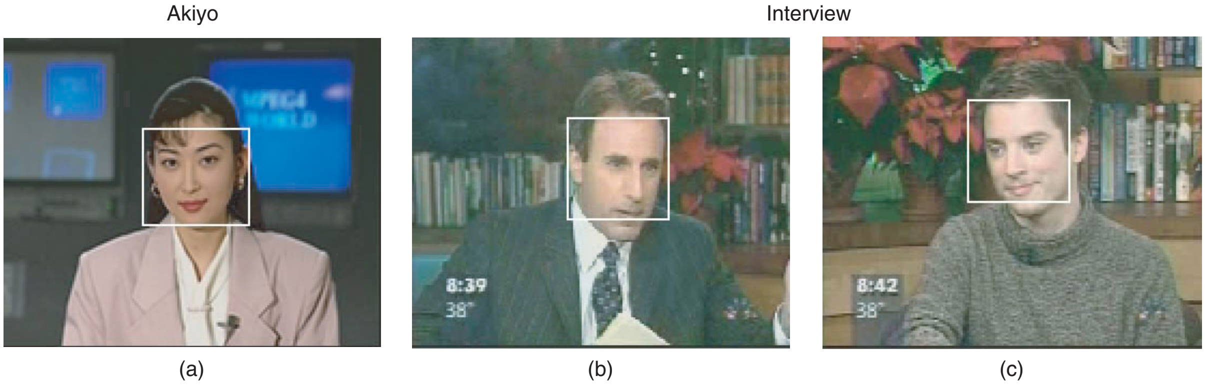

Fig. 1. Frames from sequences (a) 'Akiyo' and (b), (c) 'Interview', with objects/ROI specified.

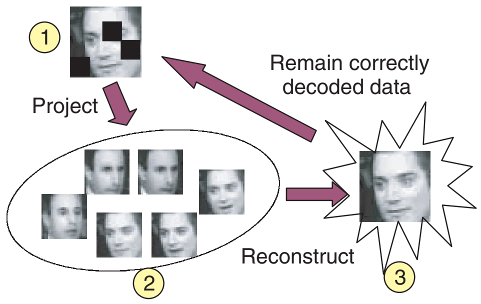

Fig. 4. POCS-based error concealment with PCA.

with PCA is shown in Figure 4. The PCA model shown in ② is pretrained with correctly received ROI. Any corrupted ROI in ① is projected to the PCA model to obtain the recovered ROI shown in ③. After the reconstructed ROI in ③ is obtained, the region in ③ in which the data was correctly received is replaced with the correctly received data. This result is projected again to the PCA model and so on until the reconstruction result is satisfactory.

In addition to PCA that can represent the ROI statistics of the training data, 'mixture of principal components' (MPC) [40] can model the multimodal characteristic of the data. 'Updating mixture of principal components' (UMPC) [11] can further adapt the model itself with the nonstationary characteristics of the ROI content. Focusing on the dynamic adapting nature of the UMPC model, we will elaborate on a simplified case of the UMPC model with a single-mixture component, which we refer to as 'updating principal components' (UPC). We will illustrate in the next session about how to use UPC as the model for second-generation error concealment to recover the lost video data.

In addition to PCA, MRF model can be used for model-based error concealment. Shirani *et al.* [41] proposed to use an appropriate form of the MRF to model the shape information of a MPEG-4 video. The MRF parameters are obtained from the edge directions of the neighbors. A maximum *a posteriori* (MAP) estimation gives the most likely reconstruction result using such an MRF model. Furthermore, model-based error concealment can use models that were originally proposed for model-based video coding. These include 3-D model-based approaches in which 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 the affine motion model. A good overview of model-based video coding can be obtained by Aizawa and Huang [42] and Pearson [43].

Temporal model-based error concealment is also possible. Models can be built for MVs. The reconstructed MVs are then used for motion compensation. For example, eigenflows proposed in Reference [44] can be used to model MVs and reconstruct any lost MV. We will detail in the next session how to update eigenflows, which we refer to as 'updating eigenflows' (U-Eigenflow), for temporal error concealment.

4. Error Concealment with Updating Principal Components (UPC)

It is important for a second-generation error-concealment method to choose a model that can represent the video content effectively. We propose to use UPC as the model for error concealment because UPC adapts itself to the nonstationary video data. Section 4.1 will describe the UPC model.

Video sequences can be either Intra or Inter coded with coding standards such as MPEG-4 [38] and H.263 [45]. The coded bitstream consists of header information, MVs, DCT coefficients, and so on. Therefore, loss of video data could be loss of any of the above information or a combination of them. In this paper, we perform error concealment for the loss of MVs and/or DCT coefficients.

In the Intra coding scenario, DCT coefficients could be lost in the transmission process. We can use UPC to reconstruct the pixel values in the ROI of a corrupted video frame. That is, spatial error concealment with UPC for pixel values in the ROI is performed (UPC for ROI). We will describe more about UPC for ROI for Intra coded videos in Section 4.2.

In the Inter coding scenario, both MVs and DCT coefficients could be lost in the transmission process. We can use UPC to reconstruct MVs of a frame. We call UPC applied to MVs, U-Eigenflow. The reconstructed MVs are then used for motion compensation. That is, U-Eigenflow for MVs is performed for a temporal error-concealment scheme to recover the lost MVs. This temporal error-concealment scheme can be used together with spatial error concealment to form a hybrid temporal/spatial error-concealment scheme. After motion compensation, UPC is further applied to pixel values in the ROI if the DCT coefficients inside this ROI are lost. The hybrid temporal/spatial error-concealment scheme with U-Eigenflow for MVs and UPC for ROI for Inter coded videos will be detailed in Section 4.3.

4.1. Updating Principal Components (UPC)

Given a set of data, we try to model the data with minimum representation error. The data given can be nonstationary, that is, the stochastic properties of the data are time-varying as shown in Figure 5(a). For example, at time instant n, the data are distributed as shown by Figure 5(a). At time instant n', the data are distributed as shown by Figure 5(b). We see that the mean of the data is shifting and that the most representative axes of the data are also rotating.

At any time instant, we attempt to represent the data as a weighted sum of the mean and the principal axes. As time proceeds, the model changes its mean and principal axes as shown in Figure 6, from Figure 6(a) and (b), so that it always models the current data effectively. To accomplish this, the representation/reconstruction error of the model evaluated at time instant n should have less contribution from the data that are further away in time from the current time instant n.

The optimization objective function at time instant n, which tries to minimize the sum of weighted reconstruction errors of all data, can be written as

$$\min_{\mathbf{m}^{(n)},\mathbf{U}^{(n)}} \sum_{i=0}^{\infty} \alpha^i \times \left\| \mathbf{x}_{n-i} - \left[\underbrace{\mathbf{m}^{(n)} + \sum_{k=1}^{P} \left[(\mathbf{x}_{n-i}-\mathbf{m}^{(n)})^{\mathrm{T}}\mathbf{u}_k^{(n)}\right]\mathbf{u}_k^{(n)}}_{\hat{\mathbf{x}}_{n-i}} \right] \right\|^2 \tag{1}$$

The notations are organized as follows:

n	:	Current time index
D	:	Dimension of the data vector
P	:	Number of eigenvectors
$\mathbf{x}_{n-i}$	:	Data vector at time $n-i$, where i represents how far away the data are from the current time instant
$\mathbf{m}^{(n)}$	:	Mean at time n
$\mathbf{u}_k^{(n)}$	:	kth eigenvector at time n
$\mathbf{U}^{(n)}$	:	Matrix with P columns of $\mathbf{u}_k^{(n)}$, $k = 1 \sim P$
$\hat{\mathbf{x}}_{n-i}$	:	Reconstruction of $\mathbf{x}_{n-i}$
α	:	Decay factor, $0 < \alpha < 1$

The reconstruction errors contributed by previous data are weighted by powers of the decay factor α. The powers are determined by how far away this sample of data is from the current time instant. At any time instant n, we try to reestimate or update the parameter (mean or eigenvector) given the parameter estimated at the previous time instant $n-1$ and the new data $\mathbf{x}_n$, by minimizing Equation (1). The solution of mean $\mathbf{m}^{(n)}$ that minimizes Equation (1) at time n is

$$\mathbf{m}^{(n)} = \alpha\mathbf{m}^{(n-1)} + (1-\alpha)\,\mathbf{x}_n \tag{2}$$

We can see that $\mathbf{m}^{(n)}$ is obtained from the previous estimated $\mathbf{m}^{(n-1)}$ and the current input $\mathbf{x}_n$. The decay factor α tells how fast the new estimation $\mathbf{m}^{(n)}$ adapts to the new data $\mathbf{x}_n$. The smaller the decay factor, the faster the estimated $\mathbf{m}^{(n)}$ adapts to the new data. Similarly, the covariance matrix $\mathbf{C}^{(n)}$ that minimizes Equation (1) at time n is

$$\mathbf{C}^{(n)} = \alpha\mathbf{C}^{(n-1)} + (1-\alpha)[(\mathbf{x}_n - \mathbf{m}^{(n)})(\mathbf{x}_n - \mathbf{m}^{(n)})^{\mathrm{T}}] \tag{3}$$

Again, $\mathbf{C}^{(n)}$ is obtained by the previous estimated $\mathbf{C}^{(n-1)}$ and the current input $\mathbf{x}_n$. The decay factor α controls how fast the eigenvectors adapt to the new data $\mathbf{x}_n$. Interested readers can read the appendix in

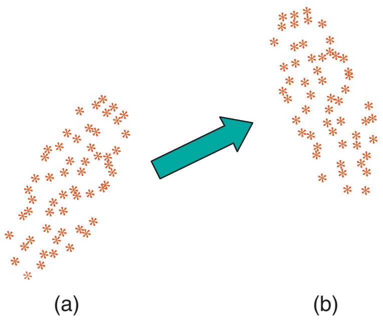

Fig. 5. Nonstationary data at (a) time n and (b) time n'.

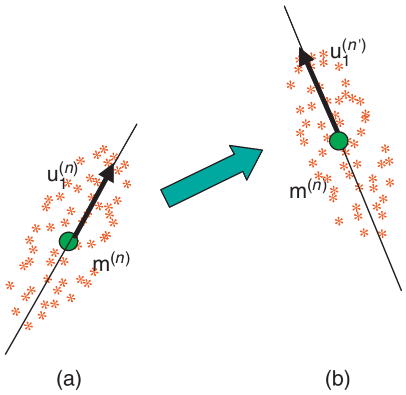

Fig. 6. UPC for nonstationary data at (a) time n and (b) time n'.

Section A1 for 'updating mixture of principal components' (UMPC), a more general case of UPC with the number of mixture components greater than one.

4.2. Spatial Error Concealment with UPC

4.2.1. UPC for ROI

As mentioned in Section 3, if ROI information is available, second-generation error concealment can be applied to pixel values in the ROI with the trained model. In this section, we consider spatial error concealment for Intra coded videos using UPC for ROI.

When the video decoder receives a video frame with an error-free ROI, it can use the data in the ROI to update the existing UPC model with the processes described in Section 4.1. In this paper, all available error-free ROI are used to update the UPC model. Less frequent update to reduce the computational complexity is possible at the expense of less adaptivity. In the experiment, the time consumed on an Intel Pentium III 650 PC to update the UPC model, with six eigenvectors, is about 100 ms per ROI. Practical system design can consider updating the UPC model with the incoming error-free ROI when the error to represent this ROI with the current UPC model is larger than a threshold.

When the video decoder receives a frame of video with corrupted macroblocks (MBs) in the ROI, it uses UPC to reconstruct the pixel values in this corrupted ROI. We adopt the POCS- based error-concealment scheme as illustrated in Figure 4. Iterations of projections and replacements are repeated until the result is satisfactory. As to reconstructing the corrupted ROI with the UPC model based on POCS, the time consumed on an Intel Pentium III 650 PC is almost negligible with 5 ms per ROI.

Error concealment for Intra coded videos is summarized in Table I.

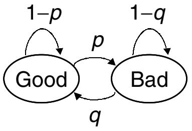

Fig. 7. Two-state Markov chain for error simulation.

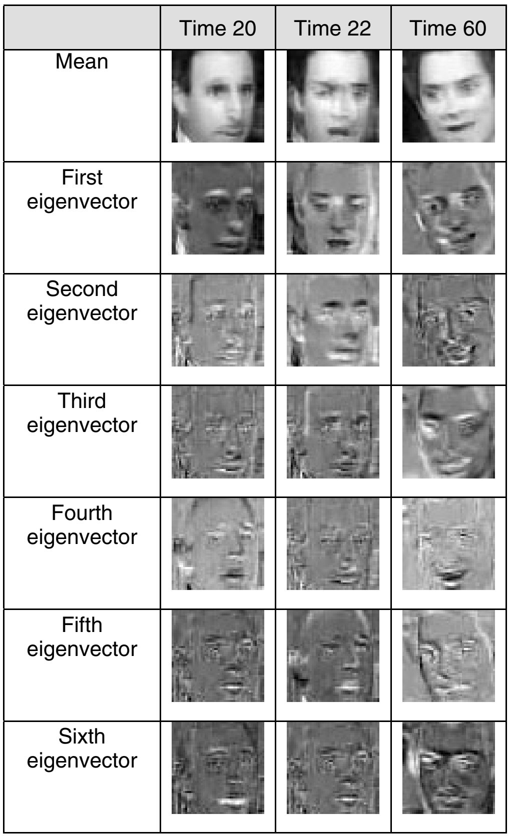

Fig. 8. Updated means and eigenvectors at time instants 20, 22, and 60.

4.2.2. Experiment

Two test video sequences 'Akiyo' and 'Interview' are used. Both video sequences are in quarter common intermediate format (QCIF). The video codec used in this paper is H.263 [45]. One sample frame of 'Akiyo' with ROI specified is shown in Figure 1(a), and two frames of 'Interview' with ROI specified are shown in Figure 1(b) and (c). Note that 'Interview' consists of two different objects of character at

Table I. Error concealment for *Intra* coded videos: spatial error concealment with UPC for ROI.

Data lost	Error concealment: training and reconstruction	
	Training for pixel values	Reconstruction
DCT coefficients only	Train UPC for ROI with ROI pixel values from frames with correctly received DCT coefficients	Apply UPC to corrupted ROI pixel values if some DCT coefficients in this ROI are lost

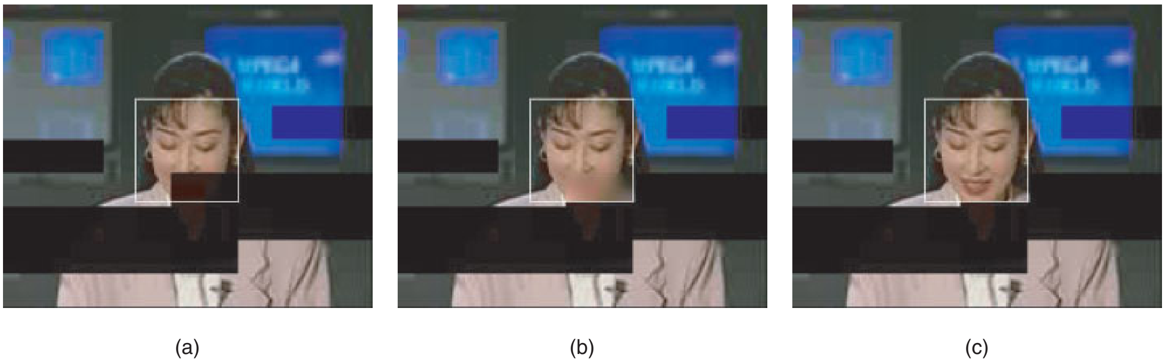

Fig. 9. Sample reconstructed frames of Intra coded 'Akiyo' with (a) no concealment; (b) concealment with spatial interpolation; or (c) concealment with UPC for ROI.

different time instances. We use a two-state Markov chain [46] to simulate the bursty error to corrupt the DCT coefficients in MBs as shown in Figure 7. 'Good' and 'Bad' correspond to error-free and erroneous states, respectively. The overall error rate ε is related to the transition probabilities p and q by $\varepsilon = p/(p+q)$. We use $\varepsilon = 0.1$ and $p = 0.01$ in the experiment. The UPC model used is with six eigenvectors, $P = 6$. In the error-concealment stage when erroneous MBs are received, five iterations of POCS are performed.

Figure 8 shows the means and eigenvectors of UPC at three different time instants 20, 22, and 60 for the sequence 'Interview'. Notice that there is a character change at time instant 21. The first character is in video frames from time 1 to 20 and the second character is in video frames from time 21 to 80. We can see that UPC describes more about the first character at time 20, as opposed to describing more about the second character at time 60. The UPC model shows a transition at time instant 22 as expected.

Figure 9 shows sample reconstructed frames of Intra coded 'Akiyo' with no concealment, spatial interpolation based on POCS, or second-generation concealment with UPC for ROI based on POCS. Spatial interpolation method interpolates the spatial neighbors of the lost MBs to recover the pixel values. The white-bounded boxes in Figure 9 are to show the ROI regions. All evaluations in peak signal-to-noise ratio (PSNR) are calculated inside the ROI region. Figure 10 shows the frame-by-frame PSNR of the three methods. Figure 11 shows sample reconstructed frames of Intra coded 'Interview' with the same three methods. Figure 12 shows the frame-by-frame PSNR of the three methods. The overall PSNR comparisons are shown in Figure 13 for both sequences.

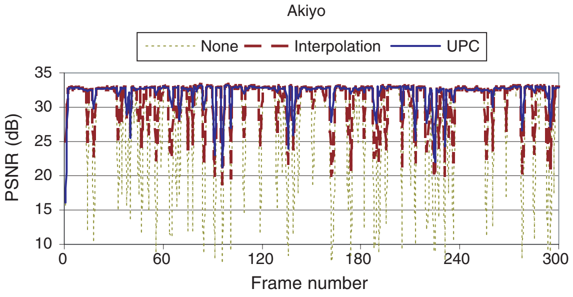

Fig. 10. Frame-by-frame PSNR of Intra coded 'Akiyo' with no concealment (none), concealment with spatial interpolation, or concealment with UPC for ROI.

We can observe that the ones without error concealment, Figures 9(a) and 11(a), have bad visual quality. The ones with spatial interpolation, Figures 9(b) and 11(b), provide some replenishment while missing the detailed texture information in the lost regions of the reconstructed ROI. Figures 9(c) and 11(c), which use UPC for ROI, provide the best reconstruction results among the three methods.

4.3. Hybrid Temporal/spatial Error Concealment with UPC

4.3.1. U-Eigenflow for MVs and UPC for ROI

In the Inter coded video bitstream, MVs as well as DCT coefficients can be lost. We apply *U-Eigenflow for MVs* to reconstruct the lost MVs for temporal error concealment. In addition, we further apply *UPC for ROI* for ROI that contain corrupted DCT coefficients. Therefore, we propose a new hybrid temporal/spatial error-concealment scheme with U-Eigenflow for MVs and UPC for ROI for Inter coded videos.

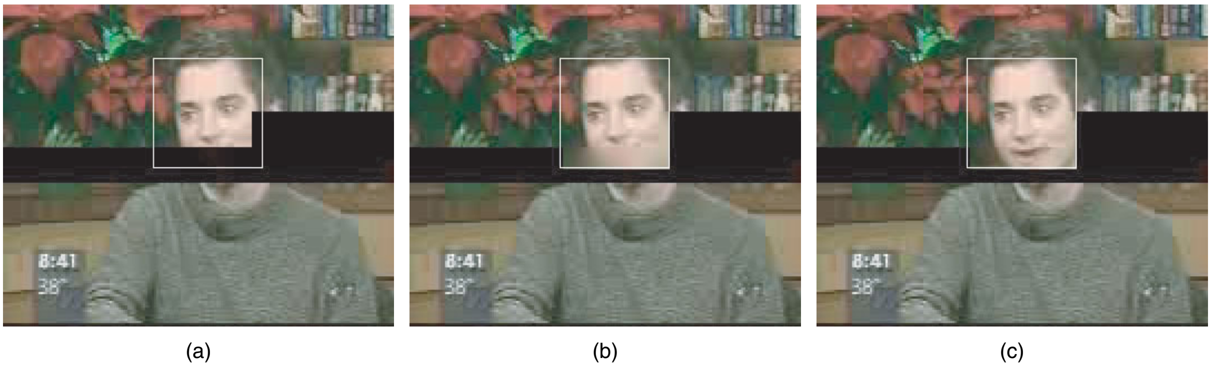

Fig. 11. Sample reconstructed frames of Intra coded 'Interview' with (a) no concealment; (b) concealment with spatial interpolation; or (c) concealment with UPC for ROI.

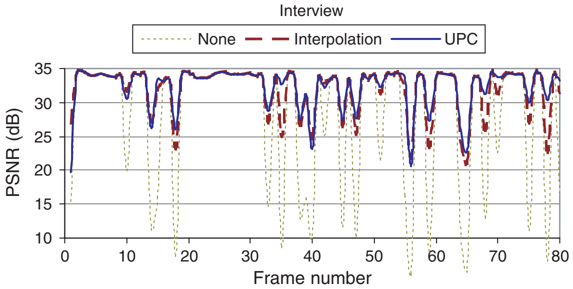

Fig. 12. Frame-by-frame PSNR of Intra coded 'Interview' with no concealment (none), concealment with spatial interpolation, or concealment with UPC for ROI.

When the video decoder receives an Intra coded video frame with error-free ROI, it uses the data in the ROI to update the existing UPC model. It is called *UPC for ROI*. When the video decoder receives an Inter coded video frame with all the MVs correct inside, it uses the MVs to update the existing U-Eigenflow. It is called *U-Eigenflow for MVs*.

When the video decoder receives a video frame with lost MVs, it uses U-Eigenflow to reconstruct the lost MVs. Motion compensation is followed using the reconstructed MVs. If no MV is lost, motion compensation uses the correctly received MVs. If there are lost DCT coefficients inside the ROI, UPC is further applied to the corrupted ROI. Both U-Eigenflow for MVs and UPC for ROI constitute the hybrid temporal/spatial error concealment for Inter coded videos.

Error concealment for Inter coded videos is summarized in Table II.

4.3.2. Experiment

The same two test video sequences 'Akiyo' and 'Interview' are used as in the Intra coded case. We use the same two-state Markov chain to simulate the bursty error to corrupt the DCT coefficients in MBs. The parameters are $\varepsilon = 0.1$ and $p = 0.01$. As to simulate the bursty error to corrupt the MVs, we use $\varepsilon = 0.05$ and $p = 0.005$ assuming that MVs are usually better protected than DCT coefficients.

U-Eigenflow for MVs and UPC for ROI both use six eigenvectors, $P = 6$. Figure 14 shows the means and the eigenvectors of U-Eigenflow at three different time instants 20, 22, and 60 for the sequence 'Interview'. Notice again that there is a character change

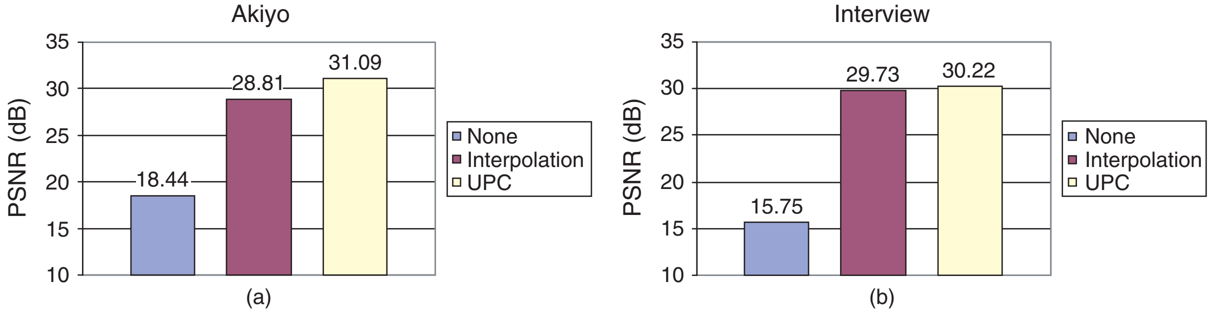

Fig. 13. Overall PSNR of Intra coded (a) 'Akiyo' and (b) 'Interview' with no concealment (none), concealment with spatial interpolation, or concealment with UPC for ROI.

Table II. Error concealment for *Inter*coded videos: hybrid temporal/spatial error concealment with U-Eigenflow for MVs/UPC for ROI.

Data lost	Error concealment: training and reconstruction	
	Training	Reconstruction
MVs in addition to DCT coefficients	1. Training for MVs—Train U-Eigenflow for MVs with correctly received MVs 2. Training for pixel values—Train UPC for ROI with ROI pixel values from Intra frames with correctly received DCT coefficients	1. Retain the lost MVs by U-Eigenflow if some MVs in a frame are lost 2. Perform motion compensation for this frame with the retained MVs from Step 1, or correctly received MVs 3. Apply UPC to ROI pixel values in the motion-compensated frame if some DCT coefficients in this ROI are lost

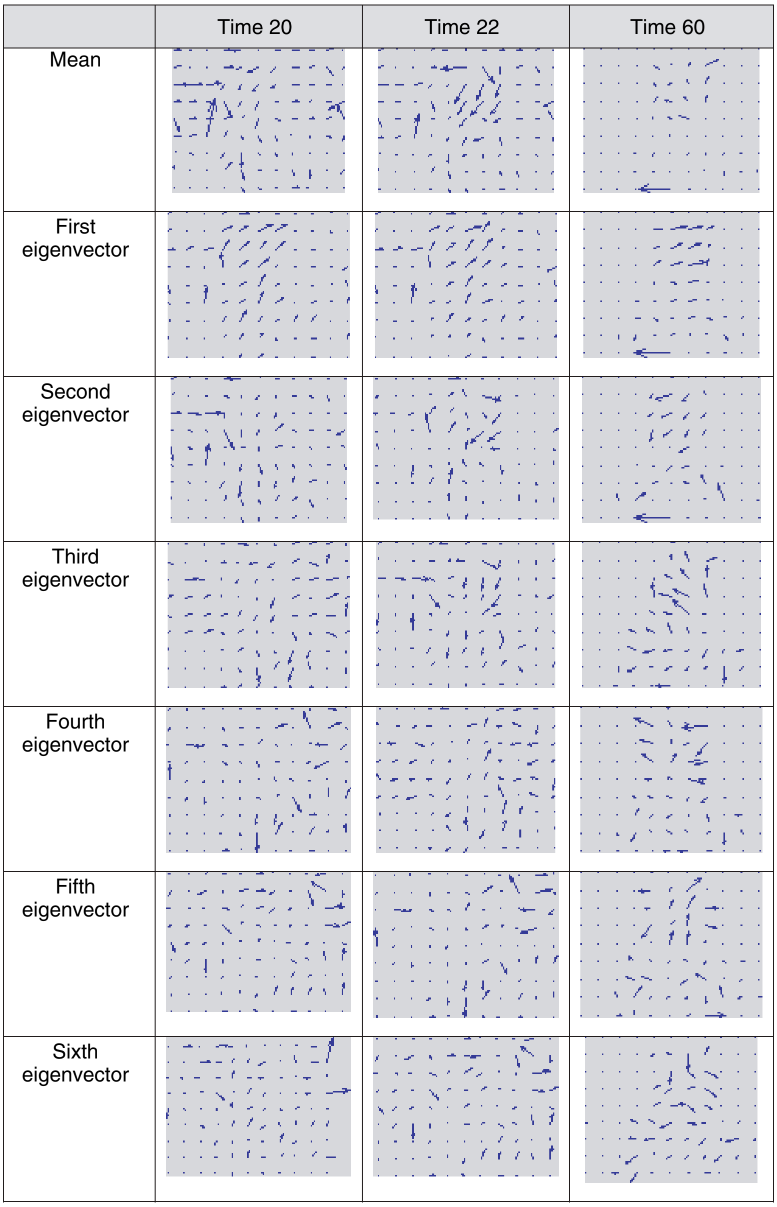

Fig. 14. Updated mean and eigenvectors of U-Eigenflow at time instants 20, 22, and 60.

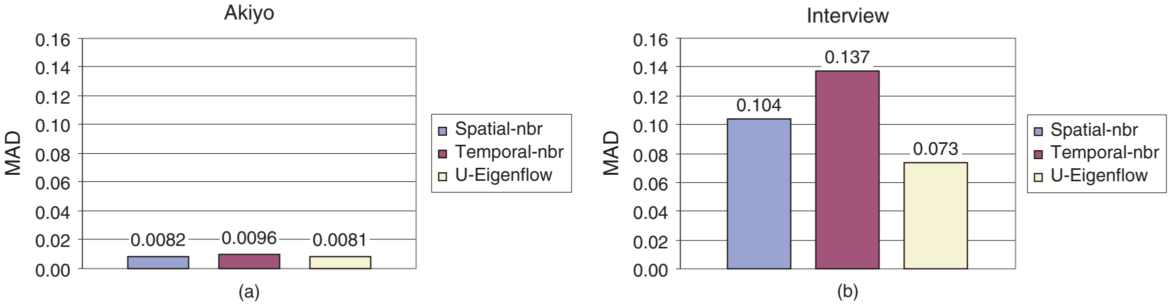

Fig. 15. MAD of reconstructed MVs of Inter coded (a) 'Akiyo' and (b) 'Interview' with spatial-nbr, temporal-nbr, or U-Eigenflow for MVs.

at time instant 21. We can see that U-Eigenflow describes more about the first character at time 20, as opposed to describing more about the second character at time 60. The first character moves his head and body a lot, while the second character moves mainly with his head. The U-Eigenflow model shows a transition at time instant 22.

The two MV reconstruction methods for comparison are 'spatial-nbr' and 'temporal-nbr'. 'Spatial-nbr' reconstructs the lost MVs for the center MB by taking the median of the eight MVs of the neighboring MBs. 'Temporal-nbr' copies the MV of the same center MB from the previous frame. After MVs have been reconstructed using these three methods, motion compensation is performed. Finally, UPC for ROI is applied to ROI pixel values if DCT coefficients in the ROI are lost.

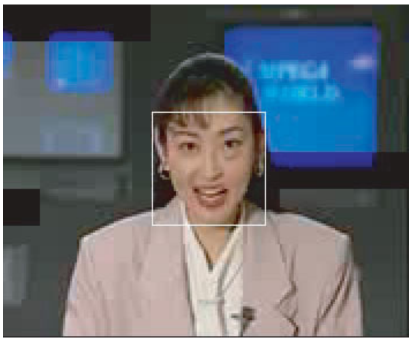

Fig. 16. Sample reconstructed frame of Inter coded 'Akiyo' with U-Eigenflow for MVs and UPC for ROI.

Figure 15(a) and (b) show the mean absolute difference (MAD) of the MVs reconstructed by these three methods for 'Akiyo' and 'Interview', respectively. 'Akiyo' has little motion throughout the sequence and all three methods can perform equally well. Figure 15(b) shows that U-Eigenflow for MVs performs the best among the three in 'Interview', in which MVs are larger.

Let us look at some sample MVs and sample reconstructed frames with the hybrid temporal/spatial error concealment. As mentioned in the last paragraph, 'Akiyo' has little motion and all three methods are equally good. Figure 16 shows the sample reconstructed frame of Inter coded 'Akiyo' with U-Eigenflow for MVs and UPC for ROI. Let us look at the results of 'Interview'. Figure 17 shows that 'temporal-nbr' and U-Eigenflow for MVs reconstruct the MVs better than 'spatial-nbr'. Figure 18 reflects the performance differences of these three methods visually. Figure 19 shows that 'spatial-nbr' and U-Eigenflow for MVs reconstruct the MVs better than 'temporal-nbr'. Figure 20 reflects the performance differences of these three methods visually. We can see that the face in Figure 20(b) has offset MBs.

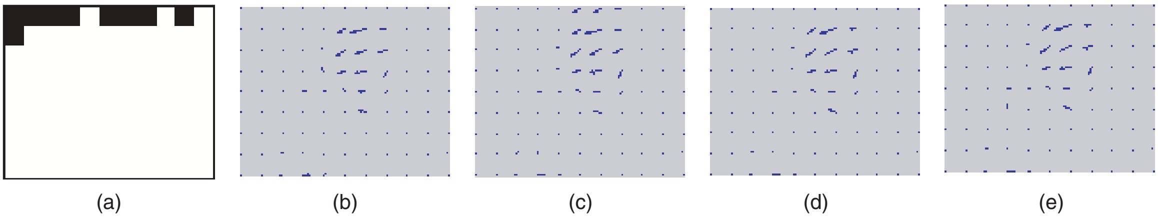

Fig. 17. Sample MVs at time 19 of Inter coded 'Interview': (a) regions indicating corrupted MVs (black blocks); (b) real MVs; (c) MVs reconstructed by spatial-nbr; (d) MVs reconstructed by temporal-nbr; and (e) MVs reconstructed by U-Eigenflow for MVs.

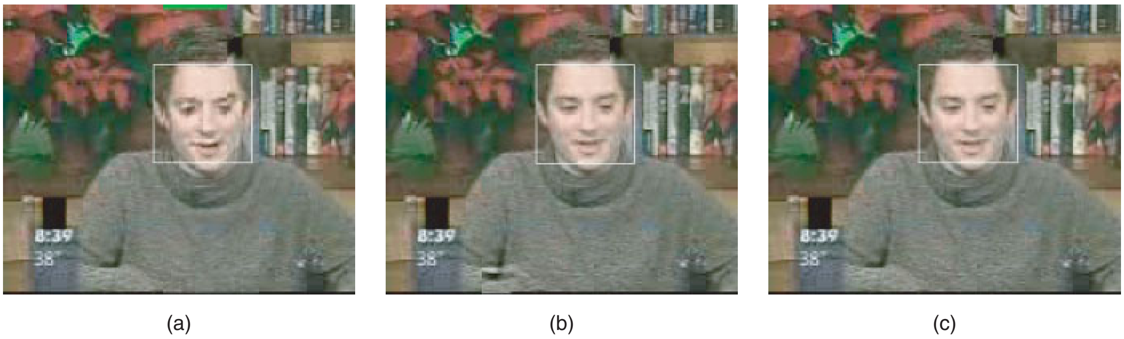

Fig. 18. Sample reconstructed frame at time 19 of Inter coded ‘Interview’ with MVs reconstructed by (a) spatial-nbr; (b) temporal-nbr; and (c) U-Eigenflow for MVs, followed by UPC for ROI for all three MV reconstruction methods.

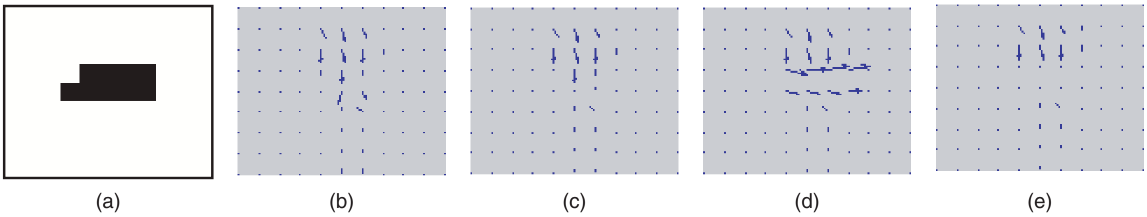

Fig. 19. Sample MVs at time 35 of Inter coded ‘Interview’: (a) regions indicating corrupted MVs (black blocks); (b) real MVs; (c) MVs reconstructed by spatial-nbr; (d) MVs reconstructed by temporal-nbr; and (e) MVs reconstructed by U-Eigenflow for MVs.

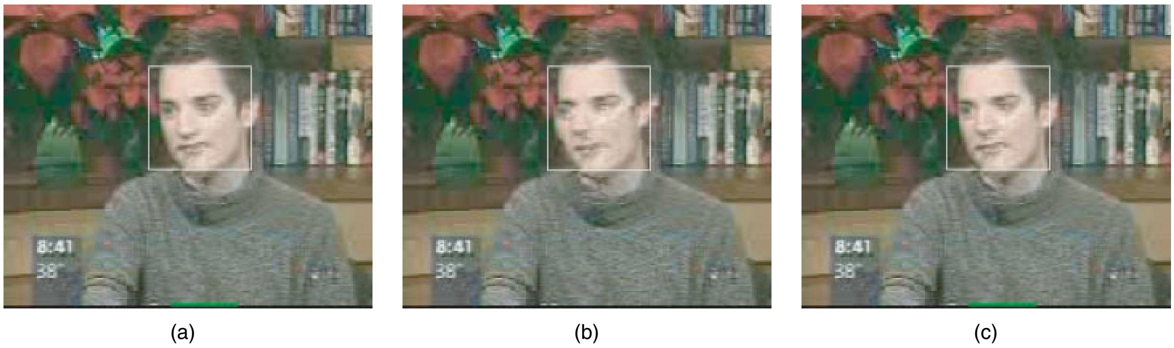

Fig. 20. Sample reconstructed frame at time 35 of Inter coded ‘Interview’ with MVs reconstructed by (a) spatial-nbr; (b) temporal-nbr; and (c) U-Eigenflow for MVs, followed by UPC for ROI for all three MV reconstruction methods.

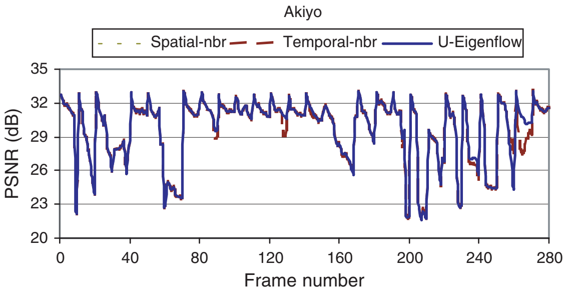

Fig. 21. Frame-by-frame PSNR of Inter coded ‘Akiyo’ with MVs reconstructed by spatial-nbr, temporal-nbr, or U-Eigenflow for MVs, followed by UPC for ROI for all three MV reconstruction methods.

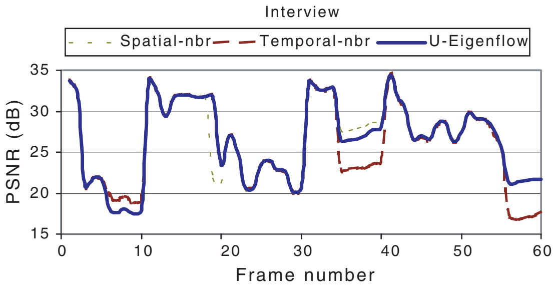

Fig. 22. Frame-by-frame PSNR of Inter coded ‘Interview’ with MVs reconstructed by spatial-nbr, temporal-nbr, or U-Eigenflow for MVs, followed by UPC for ROI for all three MV reconstruction methods.

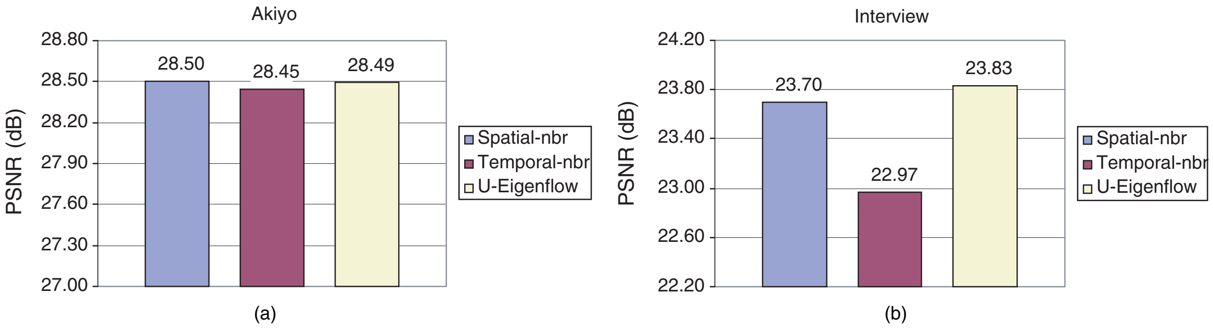

Fig. 23. Overall PSNR of Inter coded (a) 'Akiyo' and (b) 'Interview' with MVs reconstructed by spatial-nbr, temporal-nbr, or U-Eigenflow for MVs, followed by UPC for ROI for all three MV reconstruction methods.

Figures 21 and 22 show frame-by-frame PSNR performance of these three methods. The overall PSNR comparisons are shown in Figure 23 for both sequences. The hybrid error concealment with U-Eigenflow for MVs and UPC for ROI provides promising results.

5. Conclusion

In this paper, we proposed a new second-generation error-concealment framework. Second-generation error-concealment methods train and reconstruct the lost video content by context-based models and thus provide better error-concealment results than heuristic-based error-concealment methods. 'Updating principal components' (UPC) was proposed to construct such models. UPC can be applied to reconstruct the lost motion vectors (MVs) as well as to replenish the corrupted pixel values in the region of interest (ROI). The proposed hybrid temporal/spatial error concealment with U-Eigenflow for MVs and UPC for ROI provides superior performance to conventional first-generation error-concealment methods.

References

1. Wicker S. *Error Control Systems for Digital Communication and Storage*. Prentice Hall: New York, 1995.
2. Lin S, Costello DJ, Miller MJ. Automatic repeat request error control schemes. *IEEE Communication Magazine* 1984; **22**(12): 5–17.
3. Bystrom M, Parthasarathy V, Modestino JW. Hybrid error concealment schemes for broadcast video transmission over ATM networks. *IEEE Transactions on CSVT* 1999; **9**(6): 868–881.
4. Hemami SS. Robust image transmission using resynchronizing variable-length codes and error concealment. *IEEE Journal on Selected Areas in Communications* 2000; **18**(6): 927–939.
5. Aign S. Error concealment improvements for MPEG-2 using enhanced error detection and early re-synchronization. *ICASSP 1997* 1997; **4**: 2625–2628.
6. Wang Y, Zhu Q-F. Error control and concealment for video communication: a review. *Proceedings of the IEEE* 1998; **86**(5): 974–997.
7. Valente S, Dufour C, Groliere F, Snook D. An efficient error concealment implantation for MPEG-4 video streams. *IEEE Transactions on Consumer Electronics* 2001; **47**(3): 568–578.
8. Delicado F, Cuenca P, Garrido A, Orozco-Barbosa L, Quiles F. On the capabilities of error concealment in MPEG-2 communications over wireless ATM. *ICME 2000* 2000; **3**: 1443–1446.
9. Cei S, Cosman P. Comparison of error concealment strategies for MPEG video. *WCNC 1999* 1999; **1**: 329–333.
10. Turk M, Pentland A. Eigenfaces for recognition. *Journal of Cognitive Neuroscience* 1991; **3**(1): 71–86.
11. Chen TP-C, Chen T. Updating mixture of principal components for error concealment. In ICIP 2002, Rochester NY; to appear.
12. Wang Y, Zhu Q, Shaw L. Coding and cell-loss recovery in DCT based packet video. *IEEE Transactions on CSVT* 1993; **3**(3): 248–258.
13. Hemami S, Meng T. Transform coded image reconstruction exploiting interblock correlation. *IEEE Transactions on Image Processing* 1995; **4**(7): 1023–1027.
14. Suh J-W, Ho YS. Error concealment based on directional interpolation. *IEEE Transactions on Consumer Electronics* 1997; **43**(3): 295–302.
15. Zhu W, Wang Y, Zhu Q-F. Second-order derivative-based smoothness measure for error concealment. *IEEE Transactions on CSVT* 1998; **8**(6): 713–718.
16. Zeng W, Liu B. Geometric structured based error concealment with novel applications in block-based low-bit-rate coding. *IEEE Transactions on CSVT* 1999; **9**(4): 648–665.
17. Robie DL, Mersereau RM. The use of Hough transforms in spatial error concealment. *ICASSP 2000* 2000; **4**: 2131–2134.
18. Chung YJ, Kim J, Kuo CC-J. Real-time streaming video with adaptive bandwidth control and DCT-based error concealment. *IEEE Transactions on CS II: Analog and Digital Signal Processing* 1999; **46**(7): 951–956.
19. Sub H, Kwok W. Concealment of damaged block transform coded images using projections onto convex sets. *IEEE Transactions on Image Processing* 1995; **4**: 470–477.
20. Yu G-S, Liu MM-K, Marcellin MW. POCS-based error concealment for packet video using multiframe overlap information. *IEEE Transactions on CSVT* 1998; **8**(4): 422–434.
21. Geman S, Geman D. Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. *IEEE Transactions on PAMI* 1984; **PAMI-6**: 721–741.

22. Salama P, Shroff N, Delp EJ. A Bayesian approach to error concealment in encoded video streams. *ICIP 1996* 1996; **1**: 49–52.

23. Salama P, Shroff N, Delp EJ. A fast suboptimal approach to error concealment in encoded video streams. *ICIP 1997* 1997; **2**: 101–104.

24. Salama P, Shroff N, Delp EJ. Error concealment in MPEG video streams over ATM networks. *IEEE Journal on Selected Ares in Communications* 2000; **18**(6): 1129–1144.

25. Shirani S, Kossentini F, Ward R. An adaptive Markov random field based on error concealment method for video communication in an error prone environment. *ICASSP 1999* 1999; **6**: 3117–3120.

26. Zhang Y, Ma K-K. Error concealment for image transmission by multiscale Markov random field modeling. *ICME 2000* 2000; **1**: 513–516.

27. Zhang R, Regunathan SL, Rose K. Optimal estimation for error concealment in scalable video coding. *2000 Asilomar Conference on Signals, Systems and Computers* 2000; **2**: 1374–1378.

28. Lam WM, Reibman A, Liu B. Recovery of lost or erroneously received motion vectors. *ICASSP 1993* 1993; **5**: 417–420.

29. Park CS, Ye J, Lee SU. Lost motion vector recovery algorithm. *ISCAS 1994* 1994; **3**: 229–232.

30. Feng J, Lo KT, Mehrpour H, Karbowiak AE. Cell loss concealment method for MPEG video in ATM networks. *GLOBECOM 1995* 1995; **3**: 1925–1929.

31. Feng J, Lo KT, Mehrpour H. Error concealment for MPEG video transmissions. *IEEE Transactions on Consumer Electronics* 1997; **43**(2): 183–187.

32. Suh J-W, Ho YS. Motion vector recovery for error concealment based on distortion modeling. *2001 Asilomar Conference on Signals, Systems and Computers* 2001; **1**: 190–194.

33. Zhang J, Arnold JF, Frater MR, Pickering MR. Video error concealment using decoder motion vector estimation. *TENCON 1997* 1997; **2**: 777–780.

34. Zhang J, Arnold JF, Frater MR. A cell-loss concealment technique for MPEG-2 coded video. *IEEE Transactions on CSVT* 2000; **10**(4): 659–665.

35. Al-Mualla ME, Canagarajah N, Bull DR. Error concealment using motion field interpolation. *ICIP 1998* 1998; **3**: 512–516.

36. Al-Mualla ME, Canagarajah N, Bull DR. Multiple-reference temporal error concealment. *ISCAS 2001* 2001; **5**: 149–152.

37. Lee S-H, Choi D-H, Hwang C-S. Error concealment using affine transform for H.263 coded video transmissions. *Electronics Letters* 2001; **37**(4): 218–220.

38. Motion Pictures Experts Group. Overview of the MPEG-4 Standard. ISO/IEC JTC1/SC29/WG11 N2459, 1998.

39. Huang J, Chen T. Tracking of multiple faces for human-computer interfaces and virtual environments. In ICME 2000, New York, 2000.

40. Turaga DS, Chen T. Model-based error concealment for wireless video. *IEEE Transactions on CSVT* 2002; to appear.

41. Shirani S, Erol B, Kossentini F. Error concealment for MPEG-4 video communication in an error prone environment. *ICASSP 2000* 2000; **4**: 2107–2110.

42. Aizawa K, Huang TS. Model-based image coding: Advanced video coding techniques for very low bit-rate applications. *Proceedings of the IEEE* 1995; **83**: 259–271.

43. Pearson DE. Developments in model-based video coding. *Proceedings of the IEEE* 1995; **83**: 892–906.

44. Liu X, Chen T, Vijaya Kumar BVK. Face authentication for multiple subjects using eigenflow. *Pattern Recognition.* (Special Issue on Biometrics) 2001; in press.

45. ITU-T Recommendation H. 263, 27 January, 1998.

46. Yajnik M, Moon S, Kurose J, Towsley D. Measurement and modeling of the temporal dependence in packet loss. In INFOCOM 1999, March 1999; pp. 345–352.

A1. Appendix

We can extend 'updating principal components' (UPC) to more than one mixture components by modifying the objective function in Equation (1) as

$$\min_{\substack{\mathbf{m}_j^{(n)},\mathbf{U}_j^{(n)},\mathbf{w}_i\\ \forall i,j}} \left(\frac{1}{\sum_{i=0}^{\infty}\alpha^i}\right)\sum_{i=0}^{\infty}\alpha^i \left\| \mathbf{x}_{n-i} - \sum_{j=1}^{M} w_{n-i,j} \times \underbrace{\left[\mathbf{m}_j^{(n)} + \sum_{k=1}^{P}[(\mathbf{x}_{n-i}-\mathbf{m}_j^{(n)})^{\mathrm{T}}\mathbf{u}_{jk}^{(n)}]\mathbf{u}_{jk}^{(n)}\right]}_{\hat{\mathbf{x}}_{n-i,j}} \right\|^2 \quad \text{(A1)}$$

The notations are organized as follows:

n	:	Current time index
D	:	Dimension of the data vector
M	:	Number of mixture components
P	:	Number of eigenvectors in each mixture component
$\mathbf{x}_i$	:	Data vector at time i
$\mathbf{m}_j^{(n)}$	:	Mean of the jth mixture component estimated at time n
$\mathbf{u}_{jk}^{(n)}$	:	kth eigenvector of the jth mixture component estimated at time n
$\mathbf{U}_j^{(n)}$	:	Matrix with P columns of $\mathbf{u}_{jk}^{(n)}$, $k = 1 \sim P$
$\hat{\mathbf{x}}_{ij}$	:	Reconstruction of $\mathbf{x}_i$ with mixture component j
$\hat{\mathbf{X}}_i$	:	Matrix with M columns of $\hat{\mathbf{x}}_{ij}$, $j = 1 \sim M$
w_{ij}	:	Weight of $\hat{\mathbf{x}}_{ij}$ to reconstruct $\mathbf{x}_i$
$\mathbf{w}_i$	:	Vector with M entries of w_{ij}
α	:	Decay factor, $0 < \alpha < 1$
q, r	:	Index for the mixture component

The mean $\mathbf{m}_q^{(n)}$ of mixture component q at time n is

$$\mathbf{m}_q^{(n)} = \left(1 - \frac{w_{nq}^2}{\sum_{i=0}^{\infty}\alpha^i w_{n-i,q}^2}\right)\mathbf{m}_q^{(n-1)} + \left(\frac{w_{nq}}{\sum_{i=0}^{\infty}\alpha^i w_{n-i,q}^2}\right)\left(\mathbf{x}_n - \sum_{j=1,j\neq q}^{M} w_{nj}\hat{\mathbf{x}}_{nj}\right) \quad \text{(A2)}$$

The covariance matrix $\mathbf{C}_r^{(n)}$ of mixture component r at time n is

$$\begin{aligned}\mathbf{C}_r^{(n)} &= \alpha \mathbf{C}_r^{(n-1)} + (1-\alpha)\\ &\cdot \left\{ \left[(\mathbf{x}_n - w_{nr}\mathbf{m}_r^{(n)}) - \sum_{j=1, j\neq r}^{M} w_{nj}\hat{\mathbf{x}}_{nj} \right] \right.\\ &\cdot [w_{nr}(\mathbf{x}_n - \mathbf{m}_r^{(n)})]^{\mathrm{T}} + [w_{nr}(\mathbf{x}_n - \mathbf{m}_r^{(n)})]\\ &\cdot \left[(\mathbf{x}_n - w_{nr}\mathbf{m}_r^{(n)}) - \sum_{j=1, j\neq r}^{M} w_{nj}\hat{\mathbf{x}}_{nj} \right]^{\mathrm{T}}\\ &\left. - w_{nr}^2(\mathbf{x}_n - \mathbf{m}_r^{(n)})(\mathbf{x}_n - \mathbf{m}_r^{(n)})^{\mathrm{T}} \right\} \qquad \text{(A3)}\end{aligned}$$

Finally, the solution for weights is

$$\begin{bmatrix} \mathbf{w}_{n-i} \\ \lambda \end{bmatrix} = \begin{bmatrix} 2\hat{\mathbf{X}}_{n-i}^{\mathrm{T}}\hat{\mathbf{X}}_{n-i} & \mathbf{1} \\ \mathbf{1}^{\mathrm{T}} & 0 \end{bmatrix}^{-1} \begin{bmatrix} 2\hat{\mathbf{X}}_{n-i}^{\mathrm{T}}\mathbf{x}_{n-i} \\ 1 \end{bmatrix} \qquad \text{(A4)}$$

where $\mathbf{1} = [1 \cdots 1]^{\mathrm{T}}$ is an $M \times 1$ vector. The derivations of Equations (A2) to (A4) are described in the following sections.

A1.1. Solution for Updating Mixture of Principal Components

A1.1.1. Solution for the means

The optimization criterion in Equation (A1) can be rewritten as

$$\min_{\mathbf{m}_q^{(n)}} \frac{1}{\sum_{i=0}^{\infty} \alpha^i} \sum_{i=0}^{\infty} \alpha^i \|\mathbf{x}_{n-i} - \hat{\mathbf{X}}_{n-i}\mathbf{w}_{n-i}\|^2 \qquad \text{(A5)}$$

where $\hat{\mathbf{X}}_{n-i}$ is defined as

$$\begin{aligned}\hat{\mathbf{x}}_{n-i,j} &= \mathbf{U}_j^{(n)}\mathbf{U}_j^{(n)^{\mathrm{T}}}(\mathbf{x}_{n-i} - \mathbf{m}_j^{(n)}) + \mathbf{m}_j^{(n)}\\ &= \left(\underbrace{\mathbf{I}_{\mathrm{D}} - \mathbf{U}_j^{(n)}\mathbf{U}_j^{(n)^{\mathrm{T}}}}_{\mathbf{A}_j} \right) \mathbf{m}_j^{(n)} + \underbrace{\mathbf{U}_j^{(n)}\mathbf{U}_j^{(n)^{\mathrm{T}}}}_{\mathbf{B}_j} \mathbf{x}_{n-i}\end{aligned}$$

$$\hat{\mathbf{X}}_{n-i} = \sum_{j=1}^{M} \hat{\mathbf{x}}_{n-i,j}\mathbf{e}_j^{\mathrm{T}}, \text{ where}$$

$$\mathbf{e}_j = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \longleftarrow j\text{th position}$$

Note:

$$\text{Since } \mathbf{U}_j^{(n)}\mathbf{U}_j^{(n)^{\mathrm{T}}} = \mathbf{I}_M$$

$$\Rightarrow \mathbf{A}_j^{\mathrm{T}}\mathbf{A}_j = \mathbf{A}_j = \mathbf{A}_j^{\mathrm{T}} \& \mathbf{A}_j^{\mathrm{T}}\mathbf{B}_j = \mathbf{0} \qquad \text{(A6)}$$

We can see that $\mathbf{B}_j$ is the projection matrix that projects data onto a subspace spanned by the eigenvectors $\mathbf{U}_j^{(n)}$. $\mathbf{A}_j$ is the projection matrix that projects data onto a subspace that is orthogonal to the subspace spanned by the eigenvectors $\mathbf{U}_j^{(n)}$. Replacing $\hat{\mathbf{X}}_{n-i}$ in Equation (A5) with $\mathbf{A}_j$ and $\mathbf{B}_j$ in Equation (A6), we get

$$\min_{\mathbf{m}_q^{(n)}} \left(\frac{1}{\sum_{i=0}^{\infty} \alpha^i} \right) \sum_{i=0}^{\infty} \alpha^i \left\| \mathbf{x}_{n-i} - \left[\sum_{j=1}^{M} (\mathbf{A}_j\mathbf{m}_j^{(n)} + \mathbf{B}_j\mathbf{x}_{n-i})\mathbf{e}_j^{\mathrm{T}} \right] \mathbf{w}_{n-i} \right\|^2 \qquad \text{(A7)}$$

Note that $\mathbf{w}_{n-i}^{\mathrm{T}}\mathbf{e}_j = \mathbf{e}_j^{\mathrm{T}}\mathbf{w}_{n-i} = w_{n-i,j}$. Let us expand Equation (A7) and drop the terms that are independent of $\mathbf{m}_j^{(n)}$ to get

$$\begin{aligned}\min_{\mathbf{m}_q^{(n)}} \left(\frac{1}{\sum_{i=0}^{\infty} \alpha^i} \right) \sum_{i=0}^{\infty} \alpha^i\\ \times \left[\begin{aligned} &-2\sum_{j=1}^{M} w_{n-i,j}\mathbf{m}_j^{(n)^{\mathrm{T}}}\mathbf{A}_j^{\mathrm{T}}\mathbf{x}_{n-i}\\ &+\sum_{j=1}^{M}\sum_{k=1}^{M} w_{n-i,j}w_{n-i,k}\mathbf{m}_j^{(n)^{\mathrm{T}}}\mathbf{A}_j^{\mathrm{T}}\mathbf{A}_k\mathbf{m}_k\\ &+\sum_{j=1}^{M}\sum_{k=1}^{M} w_{n-i,j}w_{n-i,k}\mathbf{m}_j^{(n)^{\mathrm{T}}}\mathbf{A}_j^{\mathrm{T}}\mathbf{B}_k\mathbf{x}_{n-i}\\ &+\sum_{j=1}^{M}\sum_{k=1}^{M} w_{n-i,j}w_{n-i,k}\mathbf{x}_{n-i}^{\mathrm{T}}\mathbf{B}_j^{\mathrm{T}}\mathbf{A}_k\mathbf{m}_k^{(n)} \end{aligned} \right]\end{aligned} \qquad \text{(A8)}$$

To find the $\mathbf{m}_q^{(n)}$ that minimizes the optimization criterion, we take the derivatives of Equation (A8) with respect to $\mathbf{m}_q^{(n)}$ and set the result to zero.

$$\sum_{i=0}^{\infty}\alpha^i\left[\begin{array}{l} -w_{n-i,q}\mathbf{A}_q^{\mathrm{T}}\mathbf{x}_{n-i} \\ \quad+\displaystyle\sum_{j=1,j\neq q}^{M} w_{n-i,q}w_{n-i,j} \\ \quad\cdot\mathbf{A}_q^{\mathrm{T}}\mathbf{A}_j\mathbf{m}_j^{(n)} + w_{n-i,q}^2\mathbf{A}_q^{\mathrm{T}}\mathbf{A}_q\mathbf{m}_q^{(n)} \\ \quad+\displaystyle\sum_{k=1}^{M} w_{n-i,q}w_{n-i,k}\mathbf{A}_q^{\mathrm{T}}\mathbf{B}_k\mathbf{x}_{n-i} \end{array}\right] = \mathbf{0} \tag{A9}$$

We simplify Equation (A9) and obtain the solution as follows:

$$\mathbf{m}_q^{(n)} = \frac{1}{\displaystyle\sum_{i=0}^{\infty}\alpha^i w_{n-i,q}^2}\left[\sum_{i=0}^{\infty}\alpha^i w_{n-i,q} \cdot\left(\mathbf{x}_{n-i} - \sum_{j=1,j\neq q}^{M} w_{n-i,j}\hat{\mathbf{x}}_{n-i,j}\right)\right] \tag{A10}$$

Equation (A13) can also be written in a recursive form:

$$\mathbf{m}_q^{(n)} = \left(1 - \frac{w_{nq}^2}{\displaystyle\sum_{i=0}^{\infty}\alpha^i w_{n-i,q}^2}\right)\mathbf{m}_q^{(n-1)} + \left(\frac{w_{nq}}{\displaystyle\sum_{i=0}^{\infty}\alpha^i w_{n-i,q}^2}\right)\left(\mathbf{x}_n - \sum_{j=1,j\neq q}^{M} w_{nj}\hat{\mathbf{x}}_{nj}\right) \tag{A11}$$

A1.1.2. Solution for the eigenvectors

The optimization criterion in Equation (A1) can be rewritten as

$$\min_{\mathbf{u}_{rs}^{(n)}} \frac{1}{\displaystyle\sum_{i=0}^{\infty}\alpha^i}\sum_{i=0}^{\infty}\alpha^i\left\|\mathbf{x}_{n-i} - \sum_{j=1}^{M} w_{n-i,j} \cdot\left[\mathbf{m}_j^{(n)} + \sum_{k=1}^{P}[(\mathbf{x}_{n-i} - \mathbf{m}_j^{(n)})^{\mathrm{T}}\mathbf{u}_{jk}^{(n)}]\mathbf{u}_{jk}^{(n)}\right]\right\|^2 \tag{A12}$$

Let us expand Equation (A12) and drop the terms that are independent of $\mathbf{u}_{jk}^{(n)}$ to get

$$\min_{\mathbf{u}_{rs}^{(n)}} \frac{1}{\displaystyle\sum_{i=0}^{\infty}\alpha^i}\sum_{i=0}^{\infty}\alpha^i \cdot\left[\begin{array}{l} -2\mathbf{x}_{n-i}^{\mathrm{T}}\displaystyle\sum_{j=1}^{M} w_{n-i,j} \\ \quad\cdot\displaystyle\sum_{k=1}^{M}(\mathbf{x}_{n-i} - \mathbf{m}_j^{(n)})^{\mathrm{T}}\mathbf{u}_{jk}^{(n)}\mathbf{u}_{jk}^{(n)} \\ \quad+2\displaystyle\sum_{j=1}^{M}\sum_{a=1}^{M} w_{n-i,j}w_{n-i,a}\mathbf{m}_j^{(n)^{\mathrm{T}}} \\ \quad\cdot\displaystyle\sum_{b=1}^{P}(\mathbf{x}_{n-i} - \mathbf{m}_a^{(n)})^{\mathrm{T}}\mathbf{u}_{ab}^{(n)}\mathbf{u}_{ab}^{(n)} \\ \quad+\displaystyle\sum_{j=1}^{M}\sum_{a=1}^{M} w_{n-i,j}w_{n-i,a} \\ \quad\cdot\displaystyle\sum_{k=1}^{P}\sum_{b=1}^{P}[(\mathbf{x}_{n-i} - \mathbf{m}_j^{(n)})^{\mathrm{T}}\mathbf{u}_{jk}^{(n)}] \\ \quad\cdot[(\mathbf{x}_{n-i} - \mathbf{m}_a^{(n)})^{\mathrm{T}}\mathbf{u}_{ab}^{(n)}](\mathbf{u}_{jk}^{(n)^{\mathrm{T}}}\mathbf{u}_{ab}^{(n)}) \end{array}\right] \tag{A13}$$

Now let us simply Equation (A13) with the knowledge that eigenvectors of the same mixture component are orthogonal to each other.

$$\min_{\mathbf{u}_{rs}^{(n)}} \frac{1}{\displaystyle\sum_{i=0}^{\infty}\alpha^i}\sum_{i=0}^{\infty}\alpha^i$$

$$\cdot\begin{bmatrix} -2\sum_{j=1}^{M} w_{n-i,j}\sum_{k=1}^{M}[(\mathbf{x}_{n-i}-\mathbf{m}_j^{(n)})^{\mathrm{T}}\mathbf{u}_{jk}^{(n)}] \\ \cdot(\mathbf{x}_{n-i}^{\mathrm{T}}\mathbf{u}_{jk}^{(n)})+2\sum_{j=1}^{M}\sum_{a=1}^{M} w_{n-i,j}w_{n-i,a} \\ \cdot\sum_{b=1}^{P}[(\mathbf{x}_{n-i}-\mathbf{m}_a^{(n)})^{\mathrm{T}}\mathbf{u}_{ab}^{(n)}](\mathbf{m}_j^{(n)^{\mathrm{T}}}\mathbf{u}_{ab}^{(n)}) \\ +\sum_{j=1}^{M}\sum_{a=1,a\neq j}^{M} w_{n-i,j}w_{n-i,a} \\ \cdot\sum_{k=1}^{P}\sum_{b=1}^{P}[(\mathbf{x}_{n-i}-\mathbf{m}_j^{(n)})^{\mathrm{T}}\mathbf{u}_{jk}^{(n)}] \\ \cdot[(\mathbf{x}_{n-i}-\mathbf{m}_a^{(n)})^{\mathrm{T}}\mathbf{u}_{ab}^{(n)}](\mathbf{u}_{jk}^{(n)^{\mathrm{T}}}\mathbf{u}_{ab}^{(n)}) \\ +\sum_{j=1}^{M} w_{n-i,j}^2\sum_{k=1}^{P}[(\mathbf{x}_{n-i}-\mathbf{m}_j^{(n)})^{\mathrm{T}}\mathbf{u}_{jk}^{(n)}]^2 \end{bmatrix} \quad \text{(A14)}$$

With the additional constraint that eigenvectors need to be normal,

$$\mathbf{u}_{jk}^{(n)^{\mathrm{T}}}\mathbf{u}_{jk}^{(n)}=1 \quad \text{(A15)}$$

we can apply Lagrange optimization algorithm to find out the eigenvectors $\mathbf{u}_{rs}^{(n)}$ that minimizes the optimization criterion. Let us abbreviate the terms inside Equation (A14) as (∗), the Lagrangian function is therefore,

$$(*)+\lambda(\mathbf{u}_{jk}^{(n)^{\mathrm{T}}}\mathbf{u}_{jk}^{(n)}-1) \quad \text{(A16)}$$

Taking the derivatives of the Lagrangian function with respect to $\mathbf{u}_{rs}^{(n)}$, we get,

$$\frac{1}{\sum_{i=0}^{\infty}\alpha^i}\sum_{i=0}^{\infty}\alpha^i \cdot\begin{bmatrix} -2w_{n-i,r}[(\mathbf{x}_{n-i}-\mathbf{m}_r^{(n)})\mathbf{x}_{n-i}^{\mathrm{T}}+\mathbf{x}_{n-i}(\mathbf{x}_{n-i} \\ -\mathbf{m}_r^{(n)})^{\mathrm{T}}]\mathbf{u}_{rs}^{(n)}+2\sum_{j=1}^{M} w_{n-i,j}w_{n-i,r}[(\mathbf{x}_{n-i} \\ -\mathbf{m}_r^{(n)})\mathbf{m}_j^{(n)^{\mathrm{T}}}+\mathbf{m}_j^{(n)}(\mathbf{x}_{n-i}-\mathbf{m}_r^{(n)})^{\mathrm{T}}]\mathbf{u}_{rs}^{(n)} \\ +2\sum_{j=1,j\neq r}^{M} w_{n-i,j}w_{n-i,r}\sum_{k=1}^{P}[(\mathbf{x}_{n-i}-\mathbf{m}_j^{(n)})^{\mathrm{T}}\mathbf{u}_{jk}^{(n)}] \\ \cdot[(\mathbf{x}_{n-i}-\mathbf{m}_r^{(n)})\mathbf{u}_{jk}^{(n)^{\mathrm{T}}}+\mathbf{u}_{jk}^{(n)}(\mathbf{x}_{n-i}-\mathbf{m}_r^{(n)})^{\mathrm{T}}]\mathbf{u}_{rs}^{(n)} \\ +2w_{n-i,r}^2(\mathbf{x}_{n-i}-\mathbf{m}_r^{(n)})(\mathbf{x}_{n-i}-\mathbf{m}_r^{(n)})^{\mathrm{T}}\mathbf{u}_{rs}^{(n)} \end{bmatrix}$$

$$=-2\lambda\mathbf{u}_{rs}^{(n)} \quad \text{(A17)}$$

We can see that $\mathbf{u}_{rs}^{(n)}$ is the eigenvector of the following matrix $\mathbf{C}_r^{(n)}$:

$$\mathbf{C}_r^{(n)}=\frac{1}{\sum_{i=0}^{\infty}\alpha^i}\sum_{i=0}^{\infty}\alpha^i w_{n-i,r} \cdot\begin{bmatrix} [(\mathbf{x}_{n-i}-\mathbf{m}_r^{(n)})\mathbf{x}_{n-i}^{\mathrm{T}}+\mathbf{x}_{n-i}(\mathbf{x}_{n-i}-\mathbf{m}_r^{(n)})^{\mathrm{T}}] \\ -\sum_{j=1}^{M} w_{n-i,j}[(\mathbf{x}_{n-i}-\mathbf{m}_r^{(n)})\mathbf{m}_j^{(n)^{\mathrm{T}}} \\ +\mathbf{m}_j^{(n)}(\mathbf{x}_{n-i}-\mathbf{m}_r^{(n)})^{\mathrm{T}}]-\sum_{j=1,j\neq r}^{M} w_{n-i,j} \\ \cdot\sum_{k=1}^{P}[(\mathbf{x}_{n-i}-\mathbf{m}_j^{(n)})^{\mathrm{T}}\mathbf{u}_{jk}^{(n)}] \\ \cdot[(\mathbf{x}_{n-i}-\mathbf{m}_r^{(n)})\mathbf{u}_{jk}^{(n)^{\mathrm{T}}}+\mathbf{u}_{jk}^{(n)}(\mathbf{x}_{n-i}-\mathbf{m}_r^{(n)})^{\mathrm{T}}] \\ -w_{n-i,r}(\mathbf{x}_{n-i}-\mathbf{m}_r^{(n)})(\mathbf{x}_{n-i}-\mathbf{m}_r^{(n)})^{\mathrm{T}} \end{bmatrix} \quad \text{(A18)}$$

The first P eigenvectors of $\mathbf{C}_r^{(n)}$ are the solution for $\mathbf{u}_{rs}^{(n)}$, $s=1\sim P$. Equation (A18) can also be written in a recursive form:

$$\mathbf{C}_r^{(n)}=\alpha\mathbf{C}_r^{(n-1)}+(1-\alpha)w_{nr} \cdot\begin{bmatrix} [(\mathbf{x}_n-\mathbf{m}_r^{(n)})\mathbf{x}_n^{\mathrm{T}}+\mathbf{x}_n(\mathbf{x}_n-\mathbf{m}_r^{(n)})^{\mathrm{T}}] \\ -\sum_{j=1}^{M} w_{nj}[(\mathbf{x}_n-\mathbf{m}_r^{(n)})\mathbf{m}_j^{(n)^{\mathrm{T}}} \\ +\mathbf{m}_j^{(n)}(\mathbf{x}_n-\mathbf{m}_r^{(n)})^{\mathrm{T}}]-\sum_{j=1,j\neq r}^{M} \\ \cdot w_{nj}\sum_{k=1}^{P}[(\mathbf{x}_n-\mathbf{m}_j^{(n)})^{\mathrm{T}}\mathbf{u}_{jk}^{(n)}] \\ \cdot[(\mathbf{x}_n-\mathbf{m}_r^{(n)})\mathbf{u}_{jk}^{(n)^{\mathrm{T}}}+\mathbf{u}_{jk}^{(n)}(\mathbf{x}_n-\mathbf{m}_r^{(n)})^{\mathrm{T}}] \\ -w_{nr}(\mathbf{x}_n-\mathbf{m}_r^{(n)})(\mathbf{x}_n-\mathbf{m}_r^{(n)})^{\mathrm{T}} \end{bmatrix} \quad \text{(A19)}$$

We can rearrange Equation (A19) and get

$$\mathbf{C}_r^{(n)}=\alpha\mathbf{C}_r^{(n-1)}+(1-\alpha)$$

$$\cdot \left\{ \begin{array}{l} \left[w_{nr}(\mathbf{x}_n - \mathbf{m}_r^{(n)}) + \sum_{j=1,j\neq r}^{M} w_{nj}(\mathbf{x}_n - \hat{\mathbf{x}}_{nj}) \right] \\ \cdot [w_{nr}(\mathbf{x}_n - \mathbf{m}_r^{(n)})]^{\mathrm{T}} + [w_{nr}(\mathbf{x}_n - \mathbf{m}_r^{(n)})] \\ \cdot \left[w_{nr}(\mathbf{x}_n - \mathbf{m}_r^{(n)}) + \sum_{j=1,j\neq r}^{M} \right. \\ \left. \cdot\, w_{nj}(\mathbf{x}_n - \hat{\mathbf{x}}_{nj}) \right]^{\mathrm{T}} - w_{nr}^2(\mathbf{x}_n - \mathbf{m}_r^{(n)}) \\ \cdot (\mathbf{x}_n - \mathbf{m}_r^{(n)})^{\mathrm{T}} \end{array} \right\} \tag{A20}$$

A1.1.3. Solution for the weights

The weights are solved individually for each of the vectors $\mathbf{x}_{n-i}$. We may drop the summation over all vectors. The optimization criterion in Equation (A1) can be rewritten as

$$\min_{\mathbf{w}_{n-i}} \left\| \mathbf{x}_{n-i} - \sum_{j=1}^{M} w_{n-i,j}\hat{\mathbf{x}}_{n-i,j} \right\|^2 = \min_{\mathbf{w}_{n-i}} \|\mathbf{x}_{n-i} - \hat{\mathbf{X}}_{n-i}\mathbf{w}_{n-i}\|^2 \tag{A21}$$

There is also the constraint that the weights $\mathbf{w}_{n-i}$ for $\mathbf{x}_{n-i}$ should be summed up to one. Again, using the Lagrange optimization algorithm, we obtain Lagrangian function,

$$(\mathbf{x}_{n-i} - \hat{\mathbf{X}}_{n-i}\mathbf{w}_{n-i})^{\mathrm{T}}(\mathbf{x}_{n-i} - \hat{\mathbf{X}}_{n-i}\mathbf{w}_{n-i}) + \lambda(\mathbf{w}_{n-i}^{\mathrm{T}}\mathbf{1} - 1) \tag{A22}$$

where $\mathbf{1} = [1 \cdots 1]^{\mathrm{T}}$ is an $M \times 1$ vector. Taking the derivatives of Equation (A22) with respect to $\mathbf{w}_{n-i}$ and λ and setting the result to zero, we get,

$$\begin{bmatrix} 2\hat{\mathbf{X}}_{n-i}^{\mathrm{T}}\hat{\mathbf{X}}_{n-i} & \mathbf{1} \\ \mathbf{1}^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w}_{n-i} \\ \lambda \end{bmatrix} = \begin{bmatrix} 2\hat{\mathbf{X}}_{n-i}^{\mathrm{T}}\mathbf{x}_{n-i} \\ 1 \end{bmatrix} \tag{A23}$$

The solution for weights is therefore

$$\begin{bmatrix} \mathbf{w}_{n-i} \\ \lambda \end{bmatrix} = \begin{bmatrix} 2\hat{\mathbf{X}}_{n-i}^{\mathrm{T}}\hat{\mathbf{X}}_{n-i} & \mathbf{1} \\ \mathbf{1}^{\mathrm{T}} & 0 \end{bmatrix}^{-1} \begin{bmatrix} 2\hat{\mathbf{X}}_{n-i}^{\mathrm{T}}\mathbf{x}_{n-i} \\ 1 \end{bmatrix} \tag{A24}$$

Authors' Biographies

Trista Pei-chun Chen received the B.S. degree and the M.S. degree from National Tsing Hua University, Hsinchu, Taiwan, in 1997 and 1999, respectively. Since August 1999, she has been working towards her Ph.D. degree in Electrical and Computer Engineering at Carnegie Mellon University, Pittsburgh, Pennsylvania. From July 1998 to June 1999, she was a software engineer developing fingerprint identification algorithms at Startek Engineering Incorporated, Hsinchu, Taiwan. During the summer of 2000, she was with HP Cambridge Research Laboratory, Cambridge, Massachusetts, conducting research in image retrieval for massive databases. During the summer of 2001, she was with Pittsburgh Sony Design Center, Pittsburgh, Pennsylvania, designing circuits for Video Watermarking (VWM). Her research interests are in the areas of networked video, watermark/data hiding, image processing, and biometric signal processing. She is a student member of the IEEE.

Tsuhan Chen has been with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania, since October 1997, where he is now a Professor. He directs the Advanced Multimedia Processing Laboratory. His research interests include multimedia signal processing and communication, audio-visual interaction, biometrics, processing of 2D/3D graphics, bioinformatics, and building collaborative virtual environments. From August 1993 to October 1997, he worked in the Visual Communications Research Department, AT&T Bell Laboratories, Holmdel, New Jersey, and later at AT&T Labs-Research, Red Bank, New Jersey.

Tsuhan helped create the Technical Committee on Multimedia Signal Processing, as the founding chair, and the Multimedia Signal Processing Workshop, both in the IEEE Signal Processing Society. He has recently been appointed as the Editor-in-Chief for IEEE Transactions on Multimedia for 2002–2004. He has co-edited a book titled Advances in Multimedia: Systems, Standards, and Networks.

Tsuhan received the B.S. degree in electrical engineering from the National Taiwan University in 1987, and the M.S. and Ph.D. degrees in electrical engineering from the California Institute of Technology, Pasadena, California, in 1990 and 1993, respectively. He is a recipient of the National Science Foundation CAREER Award.