Life after Video Coding Standards: Rate Shaping and Error Concealment[*]

Trista Pei-chun Chen[1], Tsuhan Chen[1], Yuh-Feng Hsu[2]

[1] Department of Electrical and Computer Engineering, Carnegie Mellon University,
Pittsburgh, PA 15213, U.S.A.
{peichun, tsuhan}@andrew.cmu.edu
[2] Computer and Communications Research Laboratories,
Industrial Technology Research Institute,
Hsinchu 310, Taiwan
spencer@itri.org.tw

Abstract. Is there life after video coding standards? One might think that research has no room to advance with the video coding standards already defined. On the contrary, exciting research opportunities arise after the standards are specified. In this paper, we introduce two standard-related research areas: rate shaping and error concealment, as examples of interesting research that finds its context in standards. Experiment results are also shown.

1 Introduction

What are standards? Standards define a common language that different parties can communicate with each other effectively. An analogy to the video coding standard is the language. Only with the language, Shakespeare could create his work and we can appreciate the beautiful masterpiece of his. Similarly, video coding standards define the bitstream syntax, which enables the video encoder and the decoder to communicate. With the syntax and decoding procedure defined, interesting research areas such as encoder optimization, decoder post-processing, integration with the network transport and so on, are opened up. In other words, standards allow for advanced video coding research fields to be developed and coding algorithms to be compared on a common ground.

In this paper, we consider H. 263 [1] as the video coding standard example. Similar ideas can also be built on other standards such as MPEG-4 [2]. Two research areas: *rate shaping* [3] and *error concealment* [4] (Fig. 1), are introduced for networked video transport.

[*] Published in Proc. of Visual Information System (VIS'2002), Hsinchu, Taiwan, March 2002, in Lecture Notes in Computer Science series of Springer Verlag

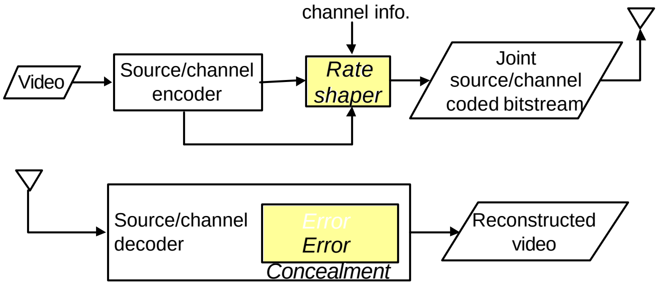


Fig. 1. System of video transport over network

First, we introduce rate shaping to perform joint source-channel coding. Video transport is very challenging given the strict bandwidth requirement and possibly high channel error rate (or packet loss rate). Through standards such as the real-time control protocol (RTCP, part of the real-time transport protocol (RTP)) [5], the encoder can obtain network condition information. The rate shaper uses such information to shape the coded video bitstream before sending it to the network. The video transport thus delivers the video bitstream with better quality and utilizes the network bandwidth more efficiently.

Second, we present error concealment with updating mixture of principle components. In a networked video application, even with good network design and video encoder, the video bitstream can be corrupted and become un-decodable at the receiver end. Error concealment is useful in such a scenario. We introduce in particular a model-based approach with updating mixture of principle components as the model. The User Datagram Protocol (UDP) [6] sequence number is used to inform the video decoder to perform error concealment.

In addition to the two areas introduced, research areas such as video traffic modeling would not be relevant without the standards being defined. Prior work on video traffic modeling can be found in [7], [8], [9], [10], and [11].

This paper is organized as follows. In Section 2, we adopt the rate shaping technique to perform joint source-channel coding. In Section 3, updating mixture of principle components is shown to perform very well in the error concealment application. We conclude this paper in Section 4.

2 Adaptive Joint Source-Channel Coding Using Rate Shaping

Video transmission is challenging in nature because it has high data rate compared to other data types/media such as text or audio. In addition, the channel bandwidth limit and error prone characteristics also impose constraints and difficulties on video transport. A joint source-channel coding approach is needed to adapt the video bitstream to different channel conditions.

We propose a joint source-channel coding scheme (Fig. 2) based on the concept of *rate shaping* to accomplish the task of video transmission. The video sequence is first source coded followed by channel coding. Popular source coding methods are H.263 [1], MPEG-4 [2], etc. Example channel coding methods are Reed-Solomon codes, BCH codes, and the recent turbo codes [12], [13]. Source coding refers to "scalable encoder/decoder" in Fig. 2 and channel coding refers to "error correction coding (ECC) encoder/decoder" in Fig. 2. The source and channel coded video bitstream then passes through the rate shaper to fit the channel bandwidth requirement while achieving the best reconstructed video quality.

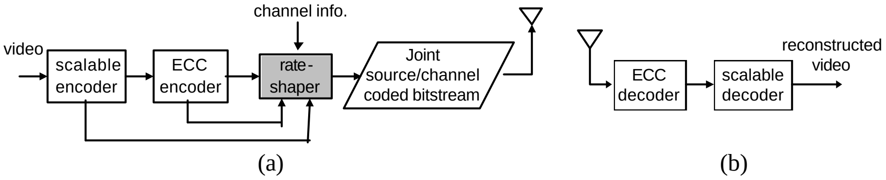


Fig. 2. System diagram of the joint source-channel coder: (a) encoder; (b) decoder

2.1 Rate Shaping

After the video sequence has been source and channel coded, the rate shaper then decides which portions of the encoded video bitstream will be sent. Let us consider the case where the video sequence is scalable coded into two layers: one base layer and one enhancement layer. Each of the two layers is error correction coded with different error correction capability. Thus, there are four *segments* in the video bitstream: the source-coding segment of the base layer bitstream (lower left segment of Fig. 3 (f)), the channel-coding segment of the base layer bitstream (lower right segment of Fig. 3 (f)), the source-coding segment of the enhancement layer bitstream (upper left segment of Fig. 3 (f)), and the channel-coding segment of the enhancement layer bitstream (upper right segment of Fig. 3 (f)). The rate shaper will decide which of the four segments to send. In the two-layer case, there are totally six valid combinations of segments (Fig. 3 (a)~(f)). We call each valid combination a *state*. Each state is represented by a pair of integers (x, y), where x is the number of source-coding segments chosen counting from the base layer and y is the number of channel-coding segments counting from the base layer. x and y satisfy the relationship of $x \geq y$.

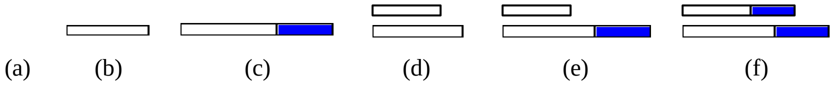
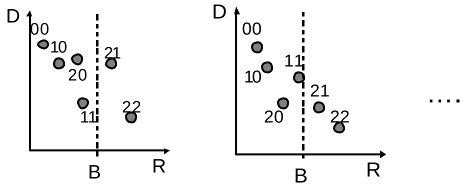


Fig. 3. Valid states: (a) State (0,0); (b) State (1,0); (c) State (1,1); (d) State (2,0); (e) State (2,1); (f) State (2,2)

The decision of the rate shaper can be optimized given the rate-distortion map, or R-D map, of each coding unit. A coding unit can be a frame, a macroblock, etc., depending on the granularity of the decision. The R-D maps vary with different channel error conditions. Given the R-D map of each coding unit with a different constellation of states (Fig. 4), the rate shaper finds the state with the minimal distortion under certain bandwidth constraint “B”. In the example of Fig. 4, State (1,1) of Unit 1 and State (2,0) of Unit 2 are chosen. Such decision is made on each of the coding unit given the bandwidth constraint “B” of that unit.



(a) (b) (c)

Fig. 4. R-D maps of coding units: (a) Unit 1; (b) Unit 2; (c) Unit 3 and so on

Consider taking a frame as a coding unit. Video bitstream is typically coded with variable bit rate in order to maintain constant video quality. To minimize the overall distortion for a group of pictures/frames (GOP), it is not enough to choose the state for each frame based on the equally allocated bandwidth to every frame. We will introduce a smart rate shaping scheme that allocates different bandwidth to each frame in a GOP. The rate shaping scheme is based on the discrete rate-distortion combination algorithm.

2.2 Discrete Rate-Distortion Combination Algorithm

Assume there are F frames in a GOP and the total bandwidth constraint for these F frames is C. Let $x(i)$ be the state chosen for frame i and let $D_{i,x(i)}$ and $R_{i,x(i)}$ be the resulting distortion and rate at frame i respectively. The goal of the rate shaper is to:

$$\text{minimize} \qquad \sum_{i=1}^{F} D_{i,x(i)} \tag{1}$$

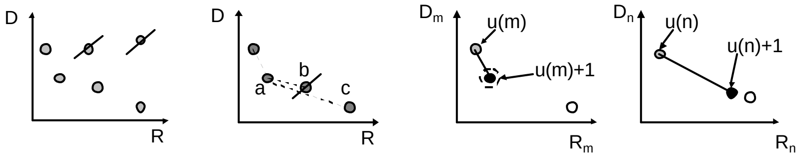
$$\text{subject to} \qquad \sum_{i=1}^{F} R_{i,x(i)} \leq C \tag{2}$$

In principle, this optimization problem can be accomplished using Dynamic Programming [14], [15], [16]. The trellis diagram is formed with the x-axis being the frame index i, y-axis being the cumulative rate at frame i, and the cost function of the trellis being the distortion. If there are S states at each frame, the number of nodes at Frame $i = F$ will be S^F (if none of the cumulative rates are the same). This method is too computationally intensive.

If the number of states, S, is large, the R-D map becomes a continuous curve. The Lagrangian Optimization method [16], [17], [18] can be used to solve this optimization problem. However, Lagrangian Optimization method cannot reach the states that do not reside on the convex hull of the R-D curve.

In this paper, we introduce a new discrete rate-distortion combination algorithm as follows:

1. At each frame, eliminate the state in the map if there exists some other state that is smaller in rate and smaller in distortion than the one considered. This corresponds to eliminating states in the upper right corner of the map (Fig. 5 (a)).
2. At each frame i, eliminate State b if $R_{ia} < R_{ib} < R_{ic}$ and $\left|\frac{D_{ib} - D_{ia}}{R_{ib} - R_{ia}}\right| < \left|\frac{D_{ic} - D_{ib}}{R_{ic} - R_{ib}}\right|$, where State a and State c are two neighboring states of State b. This corresponds to eliminating states that are on the upper right side of any line connecting two states. For example, State b is on the upper right side of the line connecting State a and State c (Fig. 5 (b)). Thus, State b is eliminated.
3. Label the remaining states in each frame from the state with the lowest rate, State 1, to the state with the highest rate. Let us denote the current decision of state at Frame i as State $u(i)$. Start from $u(i)=1$ for all frames. The rate shaper examines the next state $u(i)+1$ of each frame and finds the one that gives the largest ratio of distortion decrease over rate increase compared to the current state $u(i)$. If Frame t is chosen, increase $u(t)$ by one. As an example, let us look at two frames, Frame m and Frame n in Fig. 5 (c). Current states are represented as gray dots and the next states as black dots. We can see that updating $u(m)$ gives larger ratio increase than updating $u(n)$. Thus, the rate shaper updates $u(m)$.
4. Continue Step 3 until the total rate meets C or will exceed C with any more update of $u(i)$. If C is met, we are done.
5. If the bandwidth constraint is not yet met after Step 4, reconsider the states that were eliminated by Step 2. For each frame, re-label all the states from the state with the lowest rate to the state with the highest rate, and let $u(i)$ denote the current state. Choose the frame with the next state giving the most distortion decrease compared to the current state. If Frame t is chosen, increase $u(t)$ by one.
6. Continue Step 5 until the total rate meets C or exceeds C with more update of $u(i)$.



(a) (b) (c)

Fig. 5. Discrete R-D combination: (a) Step 1; (b) Step 2; (c) Step 3

2.3 Experiment

We compare four methods: (M1) transmits a single non-scalable and non-ECC coded video bitstream; (M2), proposed by Vass and Zhuang [19], switches between State (1, 1) and State (2, 0) depending on the channel error rate; (M3) allocates the same bit budget to each frame and chooses the state that gives the best R-D performance for each frame; (M4) is the proposed method that dynamically allocates the bit budget to each frame in a GOP and chooses the state that gives the best overall performance in a GOP, using the algorithm shown in Sect. 2.2. Each GOP has $F = 5$ frames.

The test video sequence is "stefan.yuv" in QCIF (quarter common intermediate format). The bandwidth and channel error rate vary over time and are simulated as AR(1) processes. The bandwidth ranges from 4k bits/frame to 1024k bits/frame; and the channel error rate ranges from $10^{-0.5}$ to $10^{-6.0}$.

The performance is shown in mean square error (MSE) versus the GOP number as in Fig. 6. In the case that all four methods satisfy the bandwidth constraint, the average MSE of all four methods are 10050, 5356, 2091, and 1946 respectively. The proposed M4 has the minimum distortion among all. In addition, let us compare M1 and M2 with M3 and M4. Since M1 and M2 do not have the R-D maps in mind, the network could randomly discard the bitstream sent by these two methods. The resulting MSE performance of M1 and M2 are bad. On the other hand, M3 and M4 are more intelligent in knowing that the bitstream could be non-decodable if the channel error rate is high and thus decide to allocate the bit budget to the channel-coding segments of the video bitstream.

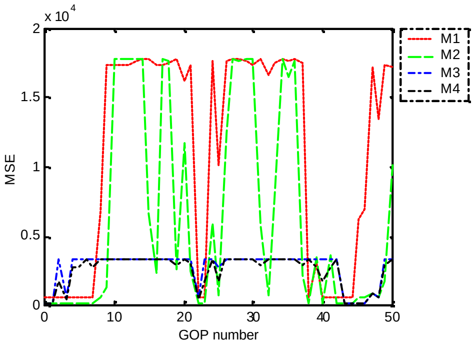


Fig. 6. MSE performance of four rate shaping methods

3 Updating Mixture of Principle Components for Error Concealment

When transmitting video data over networks, the video data could suffer from losses. Error concealment is a way to recover or conceal the loss information due to the transmission errors. Through error concealment, the reconstructed video quality can be improved at the decoder end. Projection onto convex sets (POCS) [20] is one of the most well known frameworks to perform error concealment.

Error concealment based on POCS is to formulate each constraint about the unknowns as a convex set. The optimal solution is obtained by recursively projecting a previous solution onto each convex set. For error concealment, the projections of data refer to (1) projecting the data with some losses to a model that is built on error-free data, and (2) replacing data in the loss portion with the reconstructed data. The success of a POCS algorithm relies on the model to which the data is projected onto.

We propose in this paper *updating mixture of principle components* (UMPC) to model the non-stationary as well as the multi-modal nature of the data.

It has been proposed that the *mixture of principle components* (MPC) [21] can represent the video data with a multi-modal probability distribution. For example, faces images in a video sequence can have different poses, expressions, or even changes in the characters. It is thus natural to use a multi-modal probability distribution to describe the video data. In addition, the statistics of the data may change over time as proposed by *updating principle components* (UPC) [22]. By combining the strengths of both MPC and UPC, we propose UMPC that captures both the non-stationary and the multi-modal characteristics of the data precisely.

3.1 Updating Mixture of Principle Components

Given a set of data, we try to model the data with minimum representation error. We specifically consider multi-modal data as illustrated in Fig. 7 (a). The data are clustered to multiple components (two components in this example) in a multi-dimensional space. As mentioned, the data can be non-stationary, i.e., the stochastic properties of the data are time-varying. At time n, the data are clustered as Fig. 7 (a) and at time n', the data are clustered as Fig. 7 (b). The mean of each component is shifting and the most representative axes of each component are also rotating.

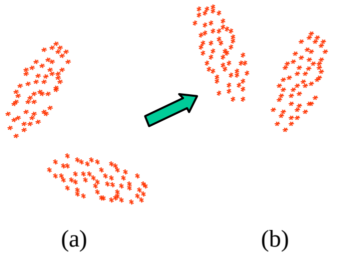


Fig. 7. Multi-modal data at (a) time n (b) time n'

At any time instant, we attempt to represent the data as a weighted sum of the mean and principle axes of each component. As time proceeds, the model changes its mean and principle axes of each component. The representation error of the model at time instant n should have less contribution from data that are further away in time from the current one. The optimization formula can be written as follows:

$$\min_{\substack{\mathbf{w}_i,\mathbf{m}_j^{(n)},\mathbf{U}_j^{(n)}\\ \forall i,j}} \left(\frac{1}{\sum_{i=0}^{\infty} a^i}\right)\sum_{i=0}^{\infty} a^i \left\| \mathbf{x}_{n-i} - \sum_{j=1}^{M} w_{n-i,j}\left[\underbrace{\mathbf{m}_j^{(n)} + \sum_{k=1}^{P}\left[\left(\mathbf{x}_{n-i} - \mathbf{m}_j^{(n)}\right)^T \mathbf{u}_{jk}^{(n)}\right]\mathbf{u}_{jk}^{(n)}}_{\hat{\mathbf{x}}_{n-i,j}}\right]\right\|^2 \tag{3}$$

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 j^{th} mixture component estimated at time n
$\mathbf{u}_{jk}^{(n)}$: k^{th} eigenvector of the j^{th} mixture component estimated at time n
$\mathbf{U}_j^{(n)}$: Matrix with P columns of $\hat{\mathbf{u}}_{jk}^{(n)}$, $k = 1 \sim P$
$\hat{\mathbf{x}}_{ij}$: 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

At any time instant n, this is to minimize the weighted reconstruction error with the choice of means, the sets of eigenvectors, and the set of weights. The reconstruction errors contributed by previous data are weighted by powers of the decay factor a. The solution to this problem is obtained by iteratively determining weights, means and sets of eigenvectors respectively while fixing the other parameters. That is, we optimize the weights for each data using the previous means and sets of eigenvectors. After updating the weights, we optimize the means and the eigenvectors accordingly. The next iteration starts again in updating the weights and so on. The iterative process is repeated until the parameters converge. At the next time instant $n+1$, the parameters of time instant n are used as the initial parameter values. Then the process of iteratively determining weights, means and sets of eigenvectors starts again.

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

$$\mathbf{m}_q^{(n)} = \left(1 - \frac{w_{nq}^2}{\sum_{i=0}^{\infty} a^i w_{n-i,q}^2} \right) \mathbf{m}_q^{(n-1)} + \left(\frac{w_{nq}}{\sum_{i=0}^{\infty} a^i w_{n-i,q}^2} \right) \left(\mathbf{x}_n - \sum_{j=1, j \neq q}^{M} w_{nj} \hat{\mathbf{x}}_{nj} \right) \tag{4}$$

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

$$\mathbf{C}_r^{(n)} = a\mathbf{C}_r^{(n-1)} + (1-a) \begin{bmatrix} w_{nr} \left[(\mathbf{x}_n - \mathbf{m}_r)\mathbf{x}_n^T + \mathbf{x}_n (\mathbf{x}_n - \mathbf{m}_r)^T \right] - \\ \sum_{j=1}^{M} w_{nj} w_{nr} \left[(\mathbf{x}_n - \mathbf{m}_r)\mathbf{m}_j^T + \mathbf{m}_j (\mathbf{x}_n - \mathbf{m}_r)^T \right] - \\ \sum_{j=1, j \neq r}^{M} w_{nj} w_{nr} \sum_{k=1}^{P} \left[\mathbf{u}_{jk}^T (\mathbf{x}_n - \mathbf{m}_j) \right] \left[(\mathbf{x}_n - \mathbf{m}_r)\mathbf{u}_{jk}^T + \mathbf{u}_{jk} (\mathbf{x}_n - \mathbf{m}_r)^T \right] - \\ w_{nr}^2 (\mathbf{x}_n - \mathbf{m}_r)(\mathbf{x}_n - \mathbf{m}_r)^T \end{bmatrix} \tag{5}$$

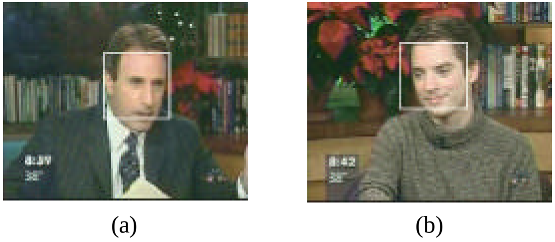
To complete one iteration with determination of means, covariance matrix and weights, the solution for weights is:

$$\begin{bmatrix} 2\hat{\mathbf{X}}_i^T \hat{\mathbf{X}}_i & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w}_i \\ l \end{bmatrix} = \begin{bmatrix} 2\hat{\mathbf{X}}_i^T \mathbf{x}_i \\ 1 \end{bmatrix} \tag{6}$$

where $\mathbf{1}=[1 \ \cdots \ 1]^T$ is an $M \times 1$ vector. We see that both MPC and UPC are special cases of UMPC with $a \rightarrow 1$ and $M = 1$ respectively.

3.2 Error Concealment with UMPC

With object based video coding standards such as MPEG-4 [2], the region of interest (ROI) information is available. A model based error concealment approach can use such ROI information and build a better error concealment mechanism. Fig. 8 shows two video frames with ROI specified. In this case, ROI can also be obtained by face trackers such as [23].


(a)

(b)

Fig. 8. Two video frames with object specified

When the video decoder receives a frame of video with error free ROI, it uses the data in ROI to update the existing UMPC with the processes described in Sect. 3.1. When the video decoder receives a frame of video with corrupted macroblocks (MB) in the ROI, it uses UMPC to reconstruct the corrupted ROI. In Fig. 9, we use three mixture components: 1st, 2nd, and 3rd, to illustrate the idea of UMPC for error concealment.

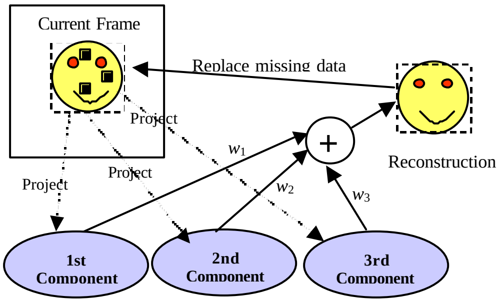


Fig. 9. UMPC for error concealment

The corrupted ROI is first reconstructed by each individual mixture component. The resulting reconstructed ROI is formed by linearly combining the three individually reconstructed ROI. The weights for linear combination are inverse proportional to the reconstruction error of each individually reconstructed ROI. After the reconstructed ROI with UMPC is done, replace the corrupted MB with the corresponding data in the reconstructed ROI just obtained. The process of reconstruction with UMPC and replacement of corrupted MB is repeated iteratively until the final reconstruction result is satisfying.

3.3 Experiment

The test video sequence is recorded from a TV program. The video codec used is H. 263 [1]. Some frames of this video sequence are shown in Fig. 8. We use a two state Markov chain [24] to simulate the bursty error to corrupt the MB as shown in Fig. 10. "Good" and "Bad" correspond to error free and erroneous state respectively. The overall error rate e is related to the transition probabilities p and q by $e = p/(p+q)$. We use $e = 0.05$ and $p = 0.01$ in the experiment.

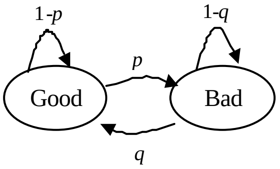


Fig. 10. Two state Markov chain for MB error simulation

There are two sets of experiments: Intra and Inter. In the Intra coded scenario, we compare three cases: (1) *none*: no error concealment takes place. When the MB is corrupted, the MB content is lost; (2) *MPC*: error concealment with MPC as the model. The number of mixture components M are three and the number of eigenvectors P for each mixture components are two; (3) *UMPC*: error concealment with UMPC as the model with $M = 3$ and $P = 2$. The decay factor is a is 0.9. In the Inter coded scenario, we also compare three cases: (1) *MC*: error concealment using motion compensation; (2) *MPC*: error concealment with MPC as the model operated on motion compensated data; (3) *UMPC*: error concealment with UMPC as the model on operated motion compensated data.

Fig. 11 shows the means of UMPC at two different time instances. It shows that the model captures three main poses of the face images. Since there is a change of characters, UMPC captures such change and we can see that the means describe more on the second character at Frame 60th.

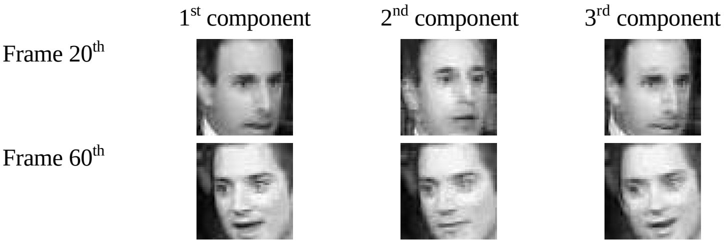


Fig. 11. Means for UMPC at Frame 20th and 60th

Fig. 12 and Fig. 13 show the decoded video frames without and with the error concealment. Fig. 12 (a) shows a complete loss of MB content when the MB data is lost. Fig. 12 (b) shows that the decoder successfully recovers the MB content with the corrupted ROI projected onto the UMPC model. Fig. 13 (a) shows the MB content being recovered by motion compensation when the MB data is lost. The face is blocky because of the error in motion compensation. Fig. 13 (b) shows that the decoder successfully recovers the MB content inside the ROI with the motion compensated ROI projected onto the UMPC model.

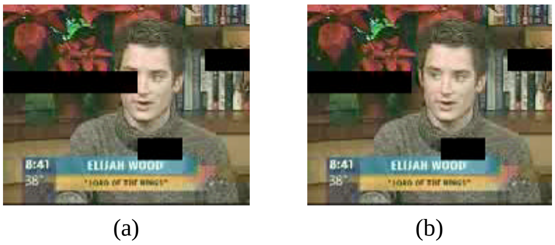


Fig. 12. Error concealment for the Intra coding scenario: (a) no concealment; (b) concealment with UMPC

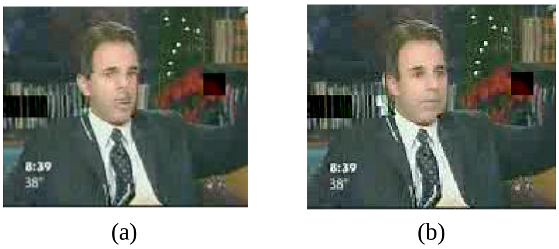


Fig. 13. Error concealment for the Inter coding scenario with: (a) motion compensation; (b) motion compensation and UMPC

The PSNR performance of the decoded video frames is summarized in Table 1. In both the Intra and Inter scenarios, error concealment with UMPC performs the best.

Table 1. Error concealment performance of four models at INTRA and INTER coded scenarios

	None (Intra) /MC (Inter)	MPC	UMPC
Intra	15.5519	29.3563	30.6657
Inter	21.4007	21.7276	22.3484

4 Conclusion

We presented two research areas: rate shaping and error concealment, that find their relevance after video coding standards are defined. With rate shaping and error concealment, we can improve the quality of service of networked video. We showed that exciting new research areas are opened up after the standards are specified.

References

1. ITU-T Recommendation H.263, January 27, 1998
2. Motion Pictures Experts Group, "Overview of the MPEG-4 Standard", ISO/IEC JTC1/SC29/WG11 N2459, 1998
3. Trista Pei-chun Chen and Tsuhan Chen, “Adaptive Joint Source-Channel Coding using Rate Shaping”, to appear in ICASSP 2002
4. Trista Pei-chun Chen and Tsuhan Chen, “Updating Mixture of Principle Components for Error Concealment”, submitted to ICIP 2002
5. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson: “RTP: A transport protocol for real-time applications”, RFC1889, Jan. 1996. ftp://ftp.isi.edu/in-notes/rfc1990.txt
6. J. Postel, “User Datagram Protocol“, RFC 768, Aug. 1980. http://www.ietf.org/rfc/rfc768.txt
7. Trista Pei-chun Chen and Tsuhan Chen, “Markov Modulated Punctured Autoregressive Processes for Traffic and Channel Modeling”, submitted to Packet Video 2002
8. D. M. Lucantoni, M. F. Neuts, and A. R. Reibman, “Method for Performance Evaluation of VBR Video Traffic Models”, IEEE/ACM Transactions on Networking, 2(2), 176-180, April 1994
9. P. R. Jelenkovic, A. A. Lazar, and N. Semret, “The Effect of Multiple Time Scales and Subexponentiality in MPEG Video Streams on Queuing Behavior”, IEEE Journal on Selected Areas in Communications, 15(6), 1052-1071
10. M. M. Krunz, A. M. Makowski, “Modeling Video Traffic using $M/G/\infty$ Input Processes: A Compromise between Markovian and LRD Models”, IEEE Journals on Selected Areas in Communications, 16(5), 733-748, 1998
11. Deepak S. Turaga and Tsuhan Chen, “Hierarchical Modeling of Variable Bit Rate Video Sources”, Packet Video 2001
12. S. Lin, D. J. Costello, Jr., Error Control Coding: Fundamentals and Application, Prentice-Hall
13. S. Wicker, Error Control Systems for Digital Communication and Storage, Prentice-Hall, 1995
14. B. Bellman, Dynamic Programming, Prentice-Hall, 1987
15. G. D. Forney, “The Viterbi Algorithm”. Proc. of the IEEE, 268-278, March 1973
16. A. Ortega and K. Ramchandran, “Rate-Distortion Methods for Image and Video Compression”. IEEE Signal Processing Magazine, 15(6), 23-50
17. H. Everett, “Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources”. Operations Research, 399-417, 1963
18. Y. Shoham and A. Gersho, “Efficient Bit Allocation for an Arbitrary Set of Quantizers”. IEEE Trans. ASSP, 1445-1453, Sep 1988
19. J. Vass and X. Zhuang, “Adaptive and Integrated Video Communication System Utilizing Novel Compression, Error Control, and Packetization Strategies for Mobile Wireless Environments”, Packet Video 2000
20. H. Sub and W. Kwok, “Concealment of Damaged Block Transform Coded Images using Projections Onto Convex Sets”, IEEE Trans. Image Processing, Vol. 4, 470-477, April 1995
21. D. S. Turaga, Ph.D. Thesis, Carnegie Mellon University, July 2001
22. X. Liu and T. Chen, "Shot Boundary Detection Using Temporal Statistics Modeling", to be appeared in ICASSP 2002
23. J. Huang and T. Chen, "Tracking of Multiple Faces for Human-Computer Interfaces and Virtual Environments", ICME 2000
24. M. Yajnik, S. Moon, J. Kurose, D. Towsley, “Measurement and modeling of the temporal dependence in packet loss”, IEEE INFOCOM, 345-52, March 1999