Streaming Video with Transformation-Based Error Concealment and Reconstruction*

Benjamin W. Wah and Xiao Su
Department of Electrical and Computer Engineering and the Coordinated Science Laboratory,
University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA,
and the Department of Computer Science and Engineering,
Chinese University of Hong Kong, Shatin, Hong Kong.
E-mail: {wah, xiao-su}@manip.crhc.uiuc.edu
URL: http://www.manip.crhc.uiuc.edu

Abstract

Real-time video streaming over the Internet requires robust delivery mechanisms with low overhead. Traditional error control schemes are not attractive because they either add redundant information that may worsen network traffic, or rely solely on the inadequate capability of the decoder to do error concealment. As sophisticated concealment techniques cannot be employed in a real-time software playback scheme, we propose in this paper a simple yet efficient transformation-based error concealment algorithm. The algorithm applies a linear transformation to the original video signals, with the objective of minimizing the mean squared error if missing information were restored by simple averaging at the destination. We also describe two strategies to cope with error propagations in temporal differentially coded frames. Experimental results show that our proposed transformation-based reconstruction algorithm performs well in real Internet tests.

Keywords and Phrases: *Internet, network loss, real-time video transmissions, reconstruction.*

1. Introduction

Real-time video streaming in the Internet is challenging because high-quality real-time transmissions cannot be sustained by the current Internet. Compression introduced to reduce the bandwidth of transmissions is often detrimental to performance when it is not designed to conceal network losses. One approach to conceal loss is to design a robust compression algorithm that can recover from losses. This is, however, difficult due to the complexity of video compression algorithms. Another approach is to design a robust transmission scheme that, under a fixed compression algorithm, allows the receiver to recover from losses. This is more practical in a real-time environment and is the approach taken in this paper.

The loss of an isolated packet may have a compound effect on subsequent dependent frames. This is caused by motion estimation and compensation algorithms employed by a video CODEC to reduce temporal redundancy. When combined with variable-length coding, the loss of one packet can result in the loss of information up to the next synchronization point in the coded bit stream, rendering subsequent packets useless even if they were received correctly.

A robust video streaming system with satisfactory quality must, therefore, have methods to deal with packet losses as well as schemes to stop error propagation.

Traditional techniques to cope with losses can be classified into *redundant transmission* and *non-redundant transmission and concealment.*

There are two strategies for adding redundancies into the transmission of coded video streams. The first strategy arranges retransmissions when packets are lost. However, it introduces a certain form of delay in the video playback.

The second strategy of adding redundancies uses error correction codes [1, 2, 7] to add redundancy into packets before transmitting them and recovers data in case of loss. These schemes consume more bandwidth and are not effective for high loss rate and bursty losses.

Non-redundant transmission and error concealment schemes [3, 4, 6, 8, 9, 11], on the other hand, recover lost pixels from those received within a tolerable range by using the inherent redundancies of source data. These schemes

*Research supported by National Science Foundation Grant MIP 96-32316 and a gift from Rockwell International.
Proc. IEEE Int'l Conf. on Multimedia Computing and Systems, 1999.

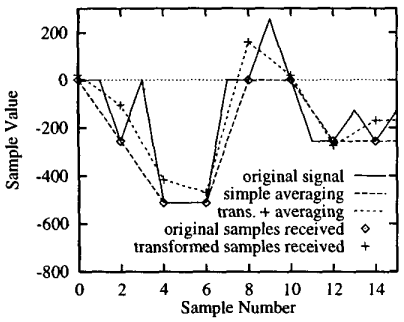


Figure 1. Reconstruction qualities between simple averaging and averaging based on transformation, assuming only the even samples are available [5].

lead to no or only slight increase in network load and is efficient when the loss rate is high. But they either assume a special delivery mechanism of the underlying network or are too complicated for real-time video playback.

We propose in this paper an efficient non-redundant transmission and concealment strategy based on interleaving and sender-side transformation.

Our proposed method is built on top of a transmission system that first interleaves pixels of video frames into streams. The interleaved streams are then compressed, packetized, and transmitted in different packets. The interleaving degree is chosen in such a way that the neighboring pixels of a missing pixel are received at the receiver with high probability. The missing pixels are reconstructed at the receiver as the average of the neighboring pixels received.

To reduce the *mean squared errors* (MSE) of all the pixels at the receiver, we transform the pixels at the sender before interleaving them. A linear transformation is designed in such a way that minimizes the MSE of all pixels when lost pixels are reconstructed by averaging. The transformation has the additional property that if all the transformed pixels are received at the receiver, the original pixels can be reconstructed.

Figure 1 illustrates the scheme that shows the original samples (in solid line), the reconstructed odd samples based on simple averaging (in dashed line), leading to SNR of 1.69 dB, and the reconstructed odd samples based on the averaging of transformed even samples (in dotted line), leading to SNR of 4.03 dB.

Two strategies are employed to reduce error propagation between frames. First, when some of the interleaved streams are received and the others lost, the reconstructed pixels are used in place of the lost pixels in motion compensation in order to have more accurate reference information. Second, pixels can be packetized in such a way that errors are not propagated from one packet to the next.

The paper is organized as follows. Section 2 discusses loss behavior in the Internet in order to determine suitable interleaving factors to be used. Section 3 overviews our video streaming system, and Section 4 presents the linear transformation algorithm that helps the receiver optimally reconstruct missing information. Finally, Section 5 shows our experimental results, and Section 6 concludes the paper.

2. Internet Loss Behavior

To design an efficient error-resilient video streaming system, we conducted a series of experiments to characterize the loss behavior of the Internet. In particular, we are interested to answer the following questions:

- Are packet losses random or bursty?
- If they are bursty, what interleaving factor should be used to make the probability of unrecoverable losses sufficiently small?

To determine the interleaving factor, we derive $p(n)$, the probability of packet losses that cannot be recovered using interleaving factor n:

$$p(n) = \frac{L - r(n)}{T} = \frac{L - (\sum_{i=1}^{n-1} iN_i + nN_n(n-1)/n)}{T} \quad (1)$$

where L denotes the total number of packets lost, T is the total number of packets received, $r(n)$ is the number of packets that can be recovered using interleaving factor n, and N_i is the frequency of i consecutive packet losses.

From our home in the Chinese University of Hong Kong (`snowdrop.cs.cuhk.edu.hk`), we chose two sites for our experiments: one between Japan (`ee.uec.ac.jp`) and Hong Kong to represent short connections, and another between USA (`math.mit.edu`) and Hong Kong to represent long-distance connections. In our experiments, we periodically sent packets (at 50 packets per second, each with 500 bytes) from the host in Hong Kong to the echo port of each of the destinations, and calculated from the packets received the probability distribution function (PDF) of burst lengths and $p(n)$, the probability of unrecoverable losses of connections to both sites. It is found that:

- Packet losses are bursty.
- Small burst lengths are sufficient. For the MIT site, the probability of not able to recover a packet is less than 0.05 with an interleaving factor of 4 and is very close to zero during its off-work hours. For the Japanese site, the failure probability is almost negligible when using an interleaving factor of 2.

Based on the above observations, we use an interleaving factor of 2 for nearby connections and 4 for transcontinental transmissions.

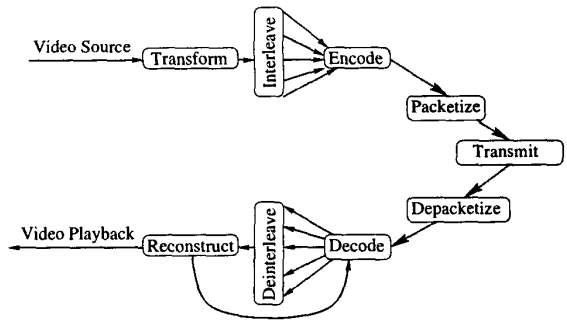


Figure 2. Components of our video streaming system

3. Overview of the Video Streaming System

Figure 2 shows the key modules of our video streaming system for transformation, interleaving/deinterleaving, compression/decompression, packetization/depacketization, and reconstruction.

We have justified our choice of the interleaving factor in Section 2 and the use of a simple reconstruction algorithm in Section 1. In the rest of this section, we present the algorithms used in compression and packetization. The transformation algorithm for minimizing reconstruction errors is presented in detail in Section 4.

3.1. Compression and Decompression

Due to the large bandwidth required for video transmissions, we need to apply an efficient video coding algorithm before sending video frames in the Internet. The ITU-T H.263 draft standard, specially designed for very low bit-rate coding (below 28.8 Kbps), is able to produce a bit rate that can be sustained in the current Internet. H.263 is largely based on the previous H.261, with a number of enhancements that can provide higher quality video at low bit rates. It uses a hybrid of transform coding and predictive coding to remove inherent spatial and temporal redundancies in video sequences.

In H.263, a video sequence is divided into segments, each consisting of an I-frame and a set of P-frames. An I-frame is intra-coded, with no dependencies on any other frame, whereas a P-frame is inter-coded through motion estimation using the previous I- or P-frame as a reference.

Since the loss of a bit stream in any frame can cause errors to propagate in the following frame, the concealment of errors at the decoder is a challenging problem. To limit the propagation of errors, we propose to feed the reconstructed frame back to the compensation loop of the H.263 decoder (also indicated in Figure 2). In our modified motion compensation algorithm, we select the reference frame for a P-frame in the following order.

1. Use the reference frame of the current interleaved stream if it is received correctly.
2. Use the reconstructed frame if any other reference frame of the same interleaving set is received intact.
3. Otherwise, use the previous reference frame of the current interleaved stream.

3.2. Packetization and Depacketization

A good packetization strategy prevents the propagations of errors among packets so that the loss of a packet will not render subsequent packets in an erroneous state.

Our packetization strategy is based on the hierarchy that H.263 organizes its bit stream. The top level is the *picture* layer that is divided into a sequence of *groups of blocks* (GOBs), each of which consists of a number of 16×16 *macroblocks* (MBs). Each MB consists of four 8×8 Y blocks, an 8×8 Cr block, and an 8×8 Cb block.

A GOB acts as the basic synchronization point in the coded stream. In most cases, when an error occurs within a GOB, the rest of the GOB will not be decodable, and the decoder has to resume synchronization at the start of the next GOB. As a result, we set our packet boundary corresponding to that of GOBs. Further, we choose our packet size of 512 bytes in order to avoid fragmentation in the Internet.

4. Transformations for Optimal Average Reconstruction

In our system, we transform input pixels in order to minimize the distortions between the original pixels and the missing ones reconstructed by averaging. For simplicity, we describe the idea based on two-way interleaving, although the idea can be extended easily to interleaving of higher degrees and other interpolation-based reconstruction methods.

4.1. Transformations for Two-way Interleaving

The sender first interleaves a video frame of size m-by-n into two streams along the horizontal direction. We describe in detail the transformations applied to each stream at the sender in order to improve the reconstruction quality in case of loss of either one of the streams.

Transformations at the sender. Suppose the original pixels of a video frame, $\vec{x} = (x_1, x_2, \ldots, x_n)^T$, are transformed into $\vec{y} = (y_1, y_2, \ldots, y_n)^T$, which is then partitioned into two sets: $\vec{y}_{even}$ containing the even numbered

pixels, and $\vec{y}_{odd}$ containing the odd numbered ones. Further, assume that only one of the sets is received at the receiver, or both sets are received.

Case I: Odd pixels received. The receiver reconstructs $\vec{y}_{even}$ by taking the average of the odd pixels received, assuming that boundaries are padded by zeroes.

$$\hat{y}_i = \begin{cases} y_i & i \text{ is odd} \\ \frac{y_{i-1}+y_{i+1}}{2} & i \text{ is even and } i \neq n \\ \frac{y_{i-1}}{2} & i = n \end{cases} \quad (2)$$

The distortion between the original and the received/reconstructed pixels is defined as:

$$D|_{\vec{y}_{odd}} = \sum_{i=1}^{n} (x_i - \hat{y}_i)^2 \quad (3)$$

$D|_{\vec{y}_{odd}}$ is minimized by setting its derivative with respect to the received y_i to be zero, where i is odd:

$$\nabla_{y_i} D|_{\vec{y}_{odd}} = 0, \qquad i = 1, 3, 5, \ldots \quad (4)$$

After simplifying (4), it can be shown that $\vec{y}_{odd}$ can be obtained by solving the following system of linear equations,

$$\vec{y}_{odd} = T\vec{x} \quad (5)$$

where T is an $\frac{n}{2}$-by-n matrix as follows:

$$T = A^{-1}B = \begin{pmatrix} 1 & \frac{1}{5} & & & & \\ \frac{1}{6} & 1 & \frac{1}{6} & & & \\ & \ddots & \ddots & \ddots & & \\ & & \frac{1}{6} & 1 & \frac{1}{6} & \\ & & & \frac{1}{6} & 1 & \end{pmatrix}^{-1} \quad (6)$$
$$\begin{pmatrix} \frac{4}{5} & \frac{2}{5} & & & & & & \\ \frac{1}{3} & \frac{2}{3} & \frac{1}{3} & & & & & \\ & \ddots & \ddots & \ddots & & & & \\ & & & \frac{1}{3} & \frac{2}{3} & \frac{1}{3} & & \\ & & & & \frac{1}{3} & \frac{2}{3} & \frac{1}{3} \end{pmatrix}$$

It has been shown in [10] that matrix A is nonsingular, so there is a unique solution to (5).

Case II: Even pixels received. Similarly, the reconstructed pixels are computed using the following:

$$\hat{y}_i = \begin{cases} y_i & i \text{ is even} \\ \frac{y_{i-1}+y_{i+1}}{2} & i \text{ is odd and } i \neq 1 \\ \frac{y_2}{2} & i = 1 \end{cases} \quad (7)$$

$\vec{y}_{even}$ is found by performing $\nabla_{y_i} D|_{\vec{y}_{even}} = 0$, where i is even. Putting the result into matrix notation yields:

$$\vec{y}_{even} = \begin{pmatrix} 1 & \frac{1}{6} & & & & \\ \frac{1}{6} & 1 & \frac{1}{6} & & & \\ & \ddots & \ddots & \ddots & & \\ & & \frac{1}{6} & 1 & \frac{1}{6} & \\ & & & \frac{1}{5} & 1 & \end{pmatrix}^{-1} \quad (8)$$
$$\begin{pmatrix} \frac{1}{3} & \frac{2}{3} & \frac{1}{3} & & & & & \\ & \frac{1}{3} & \frac{2}{3} & \frac{1}{3} & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & & \frac{1}{3} & \frac{2}{3} & \frac{1}{3} & \\ & & & & & \frac{2}{5} & \frac{4}{5} \end{pmatrix} \vec{x}$$

Case III: Both streams are received. When both interleaved streams are received, the original $\vec{x}$ can be perfectly recovered by performing the following inverse transformation.

$$T = A^{-1}B \quad (9)$$
$$= \begin{pmatrix} 1 & \frac{1}{5} & & & & \\ \frac{1}{6} & 1 & \frac{1}{6} & & & \\ & \ddots & \ddots & \ddots & & \\ & & \frac{1}{6} & 1 & \frac{1}{6} & \\ & & & \frac{1}{6} & 1 & \end{pmatrix}^{-1}$$
$$\begin{pmatrix} \frac{4}{5} & \frac{2}{5} & & & & & & \\ & \frac{1}{3} & \frac{2}{3} & \frac{1}{3} & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & & \frac{1}{3} & \frac{2}{3} & \frac{1}{3} & \\ & & & & & \frac{1}{3} & \frac{2}{3} & \frac{1}{3} \end{pmatrix}$$

4.2. Handling Burst Lengths of Four

To handle burst length of four, Our proposed transformation can be extended in two ways. First, assume that only one interleaved stream, say stream 1, is received. The remaining three streams can be restored as follows:

$$\hat{y}_{i,j} = (y_{i-1,j-1} + y_{i-1,j+1} + y_{i+1,j-1} + y_{i+1,j+1})/4 \quad (10)$$

where $y_{i,j}$ is the value of the pixel in row i and column j. The transformed values of stream 1 in order to achieve the optimal reconstruction in (10) can be derived as outlined in Section 4.1. The transformations obtained this way are overly pessimistic because they assume that three streams are always lost. In practice, it is possible for no, one, two or three streams to be lost when sent in the Internet.

Second, we can construct 4-way interleaving using a combination of 2-way interleaving as follows. We first interleave the original frame $\vec{x}$ in the horizontal direction into two streams, $\vec{x}_{h1}$ and $\vec{x}_{h2}$, and transform them. Similarly,

we perform interleaving and transformations in the vertical direction and get two additional streams. The four streams, $\vec{z}_{h1,v1}$, $\vec{z}_{h1,v2}$, $\vec{z}_{h2,v1}$ and $\vec{z}_{h2,v2}$, are then sent in distinct packets to the receiver, which carries out the following operations in order to reconstruct any missing streams.

1. If three out of the four interleaved streams are lost, say only $\vec{z}_{h1,v1}$ is received, then $\vec{z}_{h1,v2}$ can be optimally reconstructed by taking averages along the vertical direction of pixels from $\vec{z}_{h1,v1}$. By taking averages along the horizontal direction, $\vec{z}_{h2,v1}$ and $\vec{z}_{h2,v2}$ can then be recovered.
2. If two out of the four interleaved streams are lost, then there are two possible cases. If the lost streams is from the same horizontally interleaved group, say $\vec{z}_{h1,v1}$ and $\vec{z}_{h1,v2}$, then they can be optimally reconstructed by taking averages of their horizontal neighbors. If the lost streams are not from the same horizontally interleaved stream, then they can be optimally reconstructed by taking averages of their vertical neighbors.
3. If one out of the four streams is lost, then it can be reconstructed by taking averages along the vertical direction.

The strategy can be generalized to 2^m-way interleaving, for $m > 0$. It is flexible because the transformation at the sender does not depend on the loss pattern at the receiver.

5. Experimental Results

We experimented our schemes using two video sequences: *missa* (Miss America) consisting of 150 frames and *football* consisting of 90 frames. Missa represents a typical video conferencing sequence in color CIF (352 × 288) YUV format with slow head and shoulder movements, whereas football represents a fast-motion movie in YUV format with 512 × 384 resolution.

We measure the reconstructed quality by the peak signal to noise ratio (PSNR):

$$PSNR = 10 \log \frac{255^2}{\sum_i (x_i - \hat{y}_i)^2} \qquad (11)$$

where x_i and $\hat{y}_i$ are, respectively, the original and the reconstructed pixel values. In the following, we only show the PSNR values of the Y component, since it is the dominant component for human perception.

Effects of feeding back to the decoder. We modified an implementation of the H.263 decoder by Tenet RD (`http://www.nta.no/brukere/DVC/`) to incorporate feedback from the reconstruction module. Assuming two-way interleaving of the sequence into $\vec{x}_1$ and $\vec{x}_2$, and a loss pattern in which the fifth frame from $\vec{x}_1$ is always lost, Figure 3a (resp. 4a) compares the playback quality with and without feedback for the missa (resp. football) sequence. Note that from the sixth frame, the quality of the Y component without feedback can be as much as 1.8 dB lower than the one with feedback.

Tests on the Internet To compare reconstruction qualities under real situations, we tested the case with and without transformations on the Internet using our video streaming system. Both the sender and the receiver reside on the local area network in the Department of Computer Science and Engineering, Chinese University of Hong Kong. To reflect realistic Internet traffic conditions, the sender sent a stream of packets to an echo port of a remote machine, and forwarded what was returned to the receiver. The two remote sites were the ones described in Section 2.

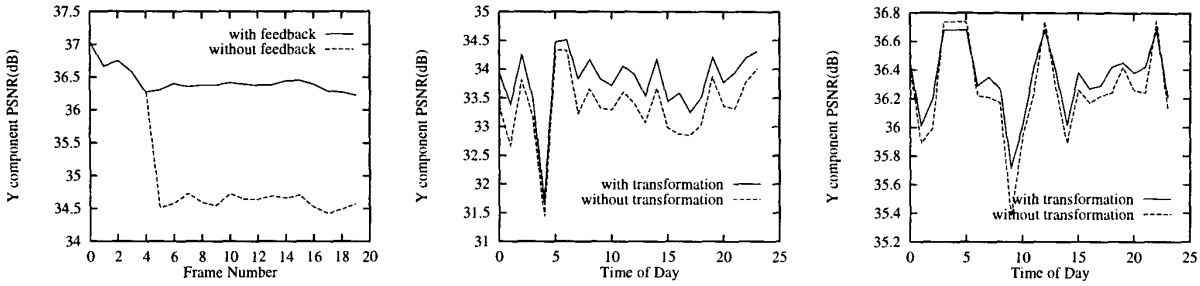
We chose a transmission rate in such a way that it did not impose excessive network load: missa sequence was sent at 30 frames/sec at a bit rate 21 kbytes/sec, and football sequence was sent at 2 frames/sec at a bit rate 22 kbytes/sec. The football sequence would have taken far more bandwidth at the normal frame rate of 30 frames/sec due to its high resolution and fast motion.

For fair comparison under the same traffic conditions, we first sent each interleaved and compressed video stream over the Internet, and recorded at the receiver loss information (the GOB number lost in transmitting each frame) in a trace file. We then ran trace-driven simulations based on the transformed and the non-transformed sequences in order to ensure that the two experiments were conducted under exactly the same Internet conditions.

Figures 3b and 4b show the results of feeding the trace of an international connection (Hong Kong – USA collected on Nov. 2, 1998) with and without transformations. For the missa sequence, our transformation-based reconstruction performs better at all times. The average PSNR (over 24 hours) is 33.80 dB if reconstruction is based on the transformed stream, and is 33.35 dB if reconstruction is based on the original stream. For the football sequence, the transformed stream gives better playback quality in most cases. The average PSNR is 24.63 dB for the transformed stream, and 24.55 dB for the original stream.

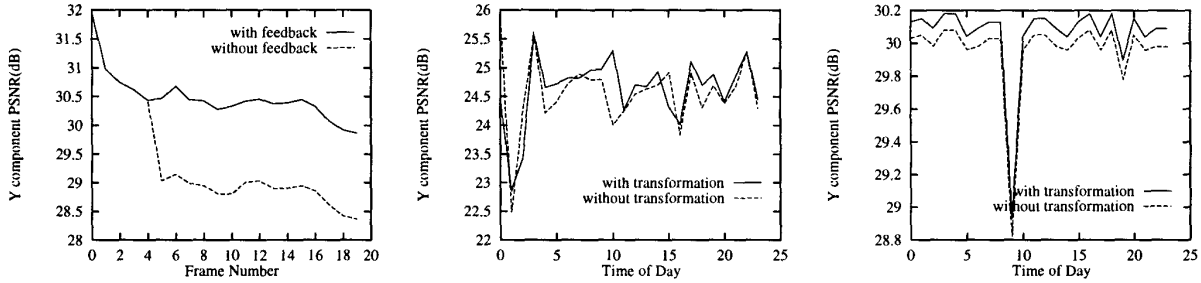
Figures 3c and 4c show the corresponding results of the Hong Kong – Japan connection. For the missa sequence, our transformation-based method performs better most of the time, with a few outliers resulted from the omission of the inverse transformation when both streams are received. The average PSNR is 36.35 dB for the transformed stream, and 36.26 dB for the original stream. For the football sequence, our transformation-based method is consistently better. The average is 30.06 dB for the transformed stream and 29.96 dB for the original stream.

In short, our results suggest that packet losses can be concealed very well in general.



(a) Effect of feedback of reconstructed frames to the motion compensation loop of decoder (b) HK-USA connection (c) HK-Japan connection

Figure 3. Results on the missa sequence.



(a) Effect of feedback of reconstructed frames to the motion compensation loop of decoder (b) HK-USA connection (c) HK-Japan connection

Figure 4. Results on the football sequence.

6. Conclusions

The paper presents a solution to cope with information loss and error propagation in real-time video streaming over the Internet. We have discussed a transformation-based signal-independent error-concealment technique that aims to minimize distortions, if some of the packets are lost and the missing information is reconstructed using simple averaging. Our experiments have shown improvements in reconstruction error at the receiver. Future research includes studies of nonlinear transformations to cope with the effects of compression and decompression.

References

[1] E. W. Biersack. Performance evaluation of forward error correction in atm networks. *Computer Communication Review*, 22(4):248–257, October 1992.

[2] J. C. Bolot and A. Vega-Garcia. Control mechanisms for packet audio in the Internet. In *Proc. IEEE Infocom'96*, pages 232–239, San Francisco, CA, April 1996.

[3] M. Ghanbari. Two-layer coding of video signals for vbr networks. *IEEE journal on Selected Areas in Communications*, 7(5):771–781, June 1989.

[4] S. S. Hemami and R. M. Gray. Subband coded image reconstruction for lossy packet networks. *IEEE Trans. on Image Processing*, April 1997.

[5] D. Lin. Real-time voice transmissions over the Internet. Master's thesis, Univ. of Illinois at Urbana-Champaign, 1999.

[6] M.Normura, T.Fujii, and N.Ohta. Layered packet-loss protection for variable rate coding using DCT. In *Proc. of International workshop on packet video*, Sept. 1988.

[7] H. Ohta and T. Kitami. A cell loss recovery method using fec in atm networks. *IEEE Journal on Selected Areas in Communications*, 9(9):1471–1483, December 1991.

[8] E. J. Posnak, S. P. Gallindo, A. P. Stephens, and H. M. Vin. Techniques for resilient transmission of jpeg video streams. In *Proc. of Multimedia Computing and Networking*, pages 243–252, San Jose, February 1995.

[9] G. Ramamurthy and D. Raychaudhuri. Performance of packet video with combined error recovery and concealment. In *Proc. of INFOCOM'95*, pages 753–761, Apr. 1995.

[10] B. W. Wah and D. Lin. Optimization based reconstruction for audio transmissions over the internet. In *Proc. 17th Symposium on Reliable Distributed Systems*. IEEE, (accepted to appear) Oct. 1998.

[11] Y. Wang and V. Ramamoorthy. Image reconstruction from partial subband images and its application in packet video transmission. *Signal Processing: Image Communication*, 3(2-3):197–229, June 1991.