Dual Frame Video Encoding with Feedback

Athanasios Leontaris and Pamela C. Cosman
Department of Electrical and Computer Engineering
University of California, San Diego, La Jolla, CA 92093-0407
Email: {pcosman,aleontar}@code.ucsd.edu, Tel: 858-822-0157 FAX: 858-822-3426

Abstract— **A special case of multiple frame prediction is the dual frame buffer, where an encoder uses both an immediate past frame and a long term past frame for motion compensation. Using a dual frame buffer together with intra/inter mode switching improves the compression performance of the coder. In this work, we investigate the effect of feedback in making better mode-switching decisions in the context of rate-distortion optimization. Feedback information is used to limit drift errors due to packet losses by synchronizing the long-term frame buffers of the encoder and decoder, and refining the input to the mode-switching decision mechanism. Experimental results show an improvement in PSNR of up to 1.6dB.**

I. Introduction

Packet-switched networks have become ubiquitous and form the backbone of the Internet. Protocols such as TCP ensure error-free packet transmission, but are not well suited for real-time delivery of streaming video content. Due to time constraints imposed by real-time operation, it is often not feasible to retransmit packets lost due to network congestion or buffer overflows. Consequently, packet losses can severely corrupt an unprotected bitstream. The bitstream has to be organized so as to minimize corruption and error propagation due to dropped packets. In this work, we assume that it is not feasible to retransmit lost packets. We will approach this problem by adopting a multiple frame prediction scheme.

The idea of using more than one past reference frame to improve coding efficiency dates a decade back [1]; it was shown that the mean-squared error (MSE) between the current frame and the predicted one decreases by using multiple frames for motion compensation. Another early attempt to code an image using a *library* of past frame components can be found in [2], and made use of vector quantization. Long-term memory multiple frame prediction was again treated in [3] in the context of a hybrid video coder coupled with rate-distortion optimized decisions over all available coding modes and reference frame indices.

Recent attemps to switch coding modes according to error robustness criteria can be found in [4], [5]. A novel algorithm for calculating estimated distortion due to packet losses was proposed in [6]. Robust video transmission within the context of long-term multiple frame prediction was studied in [7] and [8]. Feedback performance was also investigated in [7].

It quickly became apparent that multiple frame prediction leads to an often unbearable computational and memory cost. In [9], only two time-differential (reference) frames were used, thus requiring a relatively modest increase in computational complexity. We refer to this as a *dual frame buffer*. The authors showed that the scheme can have a positive impact on compression efficiency, despite using only one long-term frame. In [10] the authors use Markov chain analysis to prove that multiple frames increase error robustness. The effect of a dual frame scheme coupled with an adapted distortion estimator and rate-distortion optimized mode switching was investigated in [11].

In this paper, we use a dual frame buffer together with optimal mode switching within a rate-distortion framework as in [11], and we also use feedback to effectively synchronize the long-term frame buffers of both the encoder and decoder, and thus limit drift and error propagation due to encoder-decoder mismatch. This multiple frame prediction scheme can help the codec cope with packet losses. The paper is organized as follows. In Section II the dual frame buffer scheme is analyzed. In Section III we describe the feedback extensions, while experimental results and conclusions follow in Section IV.

II. Dual Frame Buffer

The basic idea of a dual frame buffer is as follows. While encoding frame n, the encoder and decoder both maintain two reference frames in memory. The short-term (ST) reference is frame $n-1$. The long-term (LT) reference is, say, frame $n-k$, where k may be variable, but is always greater than 1. For each macroblock (MB), there are three possible coding modes: intra, inter Short-Term (inter-ST) and inter Long-Term (inter-LT). This is illustrated in Fig. 1. The choice among them is made using rate-distortion criteria as in [11]. Once the coding mode is chosen, the syntax for encoding the bit stream is almost identical to the standard case of the single frame buffer. The only modification is that, if inter coding is chosen, a single bit will be sent to indicate use of the short-term or long-term frame. For low bit rates, this bit overhead ought to somewhat counter-balance the gain from having additional prediction options.

We now describe how the LT reference frame is chosen. In one approach, which we call *jump updating*, the LT reference frame varies from as recent as frame $n-2$ to as old as frame $n-N-1$. When encoding frame n, if the LT reference frame is $n-N-1$, then, when the encoder moves on to encoding frame $n+1$, the short-term reference frame will slide forward by one to frame n, and the LT reference frame

0-7803-8104-1/03/$17.00 ©2003 IEEE

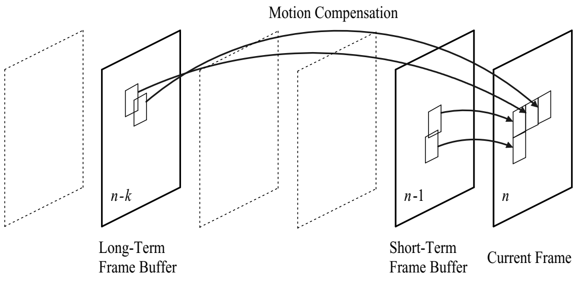

Fig. 1. Dual Frame Buffer Prediction.

will jump forward by N to frame $n-1$. The LT reference frame will then remain static for N frames, and then jump forward again. We refer to N as the jump update parameter. This approach was adopted in [9].

A novel approach, which we call *continuous updating*, entails continuously updating the long-term frame buffer so that it contains a frame with a fixed temporal distance from the current buffer. Therefore, the buffer always contains the $n-D$ frame for each frame n. We refer to D as the continuous update parameter.

We note that both jump updating and continuous updating can be viewed as special cases of a more general (N, D) updating strategy, in which the long term reference frame jumps forward by an amount N to be the frame at a distance D back from the current frame to be encoded, and then remains static for N frames, and jumps forward again. For general (N, D) updating, a frame k might have an LT frame as recent as frame $k-D$ or as old as frame $k-N-D+1$. In our definition of jump updating, N can be selected freely for each sequence, and $D=2$, (meaning that when updating occurs, the LT frame jumps forward by N to become frame $n-2$). In continuous updating, D can be selected freely for each sequence and N is fixed at 1.

The difference between the two approaches is illustrated in Fig. 2. The left side shows Jump Updating where the LT frame can be as old as 10 frames back and as recent as 2 frames back. In the top row, frame 99 is being encoded, using 98 as the ST frame and 90 as the LT frame. In the middle row, frame 100 is being encoded, and the ST frame has advanced by one to be 99, but the LT frame has remained static at 90. The LT frame is now 10 frames back, so has reached its maximum distance. In the bottom row, frame 101 is being encoded. The ST frame has again advanced by 1 to be frame 100. The LT frame has jumped forward by 9 to be frame 99.

The right side of Fig. 2 illustrates Continuous Updating. The top row also shows frame 99 being encoded using frames 98 and 90. In the next two rows, for each new frame to be encoded, both the ST and LT frames advance by one.

The most general updating strategy would have no fixed N or D; the long term frame buffer would be updated irregularly when needed, to whatever frame is most useful. In our trials, (N, D) remain fixed while coding one sequence.

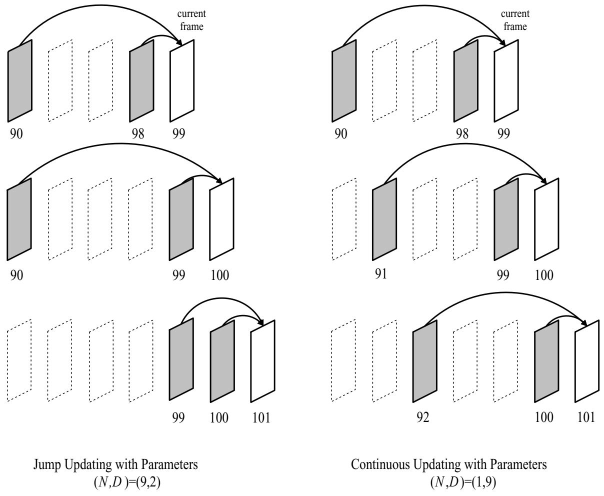

Fig. 2. Two different updating strategies for a dual frame buffer

A. Distortion Estimation

To accurately estimate distortion due to packet drops, an existing algorithm, called ROPE [6], was employed. The ROPE algorithm estimates reconstructed pixel values that incorporate potential error propagation due to packet losses. These pixel estimates are initialized at the begining of the video sequence by assuming that the first frame is always received unharmed.

We assume that the video bitstream is transmitted over a packet erasure channel. Each frame is partitioned into Groups Of Blocks (GOB). Each GOB contains a single horizontal slice of macroblocks and is transmitted as a single packet. Each packet can be independently received and decoded, due to resynchronization markers. Thus, loss of a single packet wipes out one slice of MBs, keeping the rest of the frame unharmed.

Let p be the probability of packet erasure, which is also the erasure probability for each single pixel. When the erasure is detected by the decoder, error concealment is applied. The decoder replaces the lost MB by one from the previous frame, using as motion vector (MV) the median of the MVs of the three closest MBs in the GOB above the lost one. If the GOB above has also been lost (or the 3 nearest MBs were all intra-coded and therefore have no motion vectors), then the all-zero $(0, 0)$ MV is used, and the lost MB is replaced by the co-located one from the previous frame.

Using the notation from [6], frame n of the original video signal is denoted f_n, which is compressed and reconstructed at the *encoder* as $\hat{f}_n$. The decoded (and possibly error-concealed) reconstruction of frame n at the receiver is denoted by $\tilde{f}_n$. The encoder does not know $\tilde{f}_n$, and treats it as a random variable.

Let f_n^i denote the original value of pixel i in frame n, and let $\hat{f}_n^i$ denote its *encoder* reconstruction. The reconstructed value at the *decoder*, possibly after error concealment, is denoted by $\tilde{f}_n^i$. The expected distortion for pixel i is:

$$d_n^i = E\{(f_n^i - \tilde{f}_n^i)^2\} = (f_n^i)^2 - 2f_n^i E\{\tilde{f}_n^i\} + E\{(\tilde{f}_n^i)^2\} \quad (1)$$

Calculation of d_n^i requires the first and second moments of the random variable of the estimated image sequence $\tilde{f}_n^i$. To compute these, recursion functions are developed in [6], in which it is necessary to separate out the cases of intra- and inter-coded MBs. In our approach, however, we have two separate inter modes, the *inter-ST* and *inter-LT*. Let i denote the pixel in the current frame, k denote the pixel in the previous frame that is associated with pixel i in the current frame using error concealment. Finally, j denotes the pixel in the reference frame (either ST or LT) that is the prediction of pixel i in the current frame derived using the motion vector.

The two required moments for a pixel in an *intra*-coded MB are [6]:

$$E\{\tilde{f}_n^i\} = (1-p)(\hat{f}_n^i) + p(1-p)E\{\tilde{f}_{n-1}^k\} + p^2 E\{\tilde{f}_{n-1}^i\} \quad (2)$$

$$\begin{aligned} E\{(\tilde{f}_n^i)^2\} &= (1-p)(\hat{f}_n^i)^2 + p(1-p)E\{(\tilde{f}_{n-1}^k)^2\} \\ &+ p^2 E\{(\tilde{f}_{n-1}^i)^2\} \end{aligned} \quad (3)$$

Identical to the inter case in [6], the first and second moments of $\tilde{f}_n^i$ for a pixel in an *inter-ST*-coded MB are:

$$\begin{aligned} E\{\tilde{f}_n^i\} &= (1-p)\left(\hat{e}_n^i + E\{\tilde{f}_{n-1}^j\}\right) + p(1-p)E\{\tilde{f}_{n-1}^k\} \\ &+ p^2 E\{\tilde{f}_{n-1}^i\} \end{aligned} \quad (4)$$

$$\begin{aligned} E\{(\tilde{f}_n^i)^2\} &= (1-p)\Big((\hat{e}_n^i)^2 + 2\hat{e}_n^i E\{\tilde{f}_{n-1}^j\} \\ &+ E\{(\tilde{f}_{n-1}^j)^2\}\Big) + p(1-p)E\{(\tilde{f}_{n-1}^k)^2\} \\ &+ p^2 E\{(\tilde{f}_{n-1}^i)^2\} \end{aligned} \quad (5)$$

Finally, for an *inter-LT*-coded MB we obtain:

$$\begin{aligned} E\{\tilde{f}_n^i\} &= (1-p)\left(\hat{e}_n^i + E\{\tilde{f}_{n-l}^j\}\right) + p(1-p)E\{\tilde{f}_{n-1}^k\} \\ &+ p^2 E\{\tilde{f}_{n-1}^i\} \end{aligned} \quad (6)$$

$$\begin{aligned} E\{(\tilde{f}_n^i)^2\} &= (1-p)\Big((\hat{e}_n^i)^2 + 2\hat{e}_n^i E\{\tilde{f}_{n-l}^j\} \\ &+ E\{(\tilde{f}_{n-l}^j)^2\}\Big) + p(1-p)E\{(\tilde{f}_{n-1}^k)^2\} \\ &+ p^2 E\{(\tilde{f}_{n-1}^i)^2\} \end{aligned} \quad (7)$$

B. Rate-Distortion Optimization

The encoder switches between intra, inter-ST or inter-LT coding on a macroblock basis, in an optimal fashion for a given bit rate and packet loss rate. The goal is to minimize the total distortion subject to a bit rate constraint. Individual macroblock contributions to this cost are additive, thus it can be minimized on a macroblock basis. Therefore, the encoding mode for each MB is chosen by minimizing

$$\min_{(mode,QP)} J_{MB} = \min_{(mode,QP)} (D_{MB} + \lambda R_{MB}) \quad (8)$$

D_{MB} and R_{MB} denote per MB distortion and rate, respectively. λ is the Lagrange multiplier. Both the *coding mode* (intra, inter-ST and inter-LT) and the *quantization step size* QP (ranges from 1 to 31) are chosen to minimize the Lagrangian cost. Thus, the search for optimal coding parameters is conducted over 93 combinations, compared to 62 for the single-frame case, yielding an increase in complexity of 50%.

III. Feedback Extensions

Experimental results in [6] for the single frame case showed that the intelligent use of feedback information (acknowledgement of received packets) can lead to substantial improvements in performance. We now describe the use of feedback for the dual frame encoder.

A. System Description

Let i be the current frame's index. Using feedback with a fixed delay d, the encoder can have perfect knowledge of the decoder's $(i-d)$-th reconstructed frame. We use the term "re-decode" to describe the encoder's process of using the feedback information to decode a past frame so that it is identical to the decoder's version of that frame. As the encoder knows which GOBs were received intact and which ones were dropped, it can simulate the decoder's operation exactly, including error concealment. We use the term "estimate" to describe a frame at the encoder for which the feedback information is not yet available, so the encoder is forced to estimate the decoder version. The estimate is based on using the packet loss probability and the decoder's error concealment method to estimate the distortion of each pixel in the decoder's frame, including error propagation [6].

One approach to using feedback is to make the LT frame buffer move forward to contain the closest *exactly known* frame, that is the $(i-d)$ frame. The feedback allows us to improve the estimate of the ST frame, and reduce the estimation error for the LT frame to zero. We ensure that the encoder and decoder LT frame buffers always contain an *identical* reconstruction. An example of this approach for $N=2$ and $d=5$ is depicted in Fig. 3. In Fig. 3, frame 12 is currently being encoded. Its LT frame is frame 7 which has been re-decoded. However, re-decoding frame 7 required the re-decoded versions of frames 1 and 6, its ST and LT frames, respectively. Now we can obtain the estimates of 8, 9, 10 and 11. For frame 8, 7 and 3 (both re-decoded) will be required. For 9 we will need *estimated* 8 (ST) and re-decoded 3 (LT). For 10 we will need estimated 9 and re-decoded 5. Similarly, 11 needs estimated 10 and re-decoded 5.

By synchronizing the long-term frame buffers at the transmitter and receiver, we totally eliminate drift errors: inter-LT

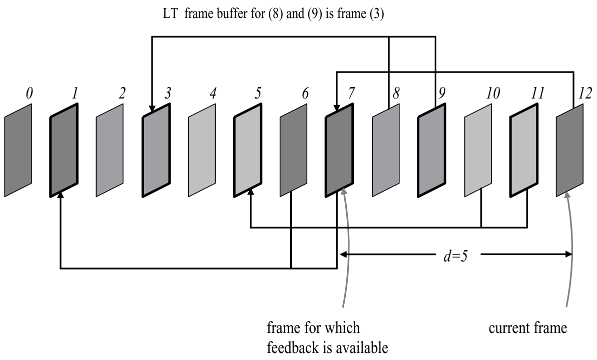

Fig. 3. Example of feedback where $N = 2$, $D = 5$ and $d = 5$.

encoded macroblocks, if they arrive, will be reconstructed in an identical manner at the encoder and decoder. Normally, this is only guaranteed by transmitting intra-coded macroblocks. Here, however, feedback signals enable us to use the long-term frame buffer as an additional error robustness factor without sacrificing greatly in compression efficiency.

This is a major difference from using the original ROPE estimator [6] with feedback. Instead of using feedback only to improve the distortion estimate and therefore the mode selection, we now, in addition, use this information to re-decode the LT frame at the encoder and thus improve motion estimation, by using a more realistic reference frame. As we will see, the codec performs very well under a variety of conditions.

B. Buffering Requirements

The feedback approach requires buffering additional frames at the encoder, beyond just the ST and LT reference frames. When feedback information arrives for frame $i - d$, the encoder must have saved the ST and LT frames for it in order to re-decode it. Likewise, to update the estimates for the frames between frame $i - d$ and the current one, the encoder must save ST and LT frames for them.

For example, consider the encoding of frame 12 in Fig. 3. Frame 7 has just been re-decoded, and we wish to use this re-decoded frame to improve the estimates of frames 8, 9, 10, and 11. First of all, re-decoded frames 1 and 6 must have been buffered in order to re-decode frame 7, as they were the LT and ST frames for frame 7. After re-decoding frame 7, the encoder can purge re-decoded frames 1 and 6, since these will no longer be needed. However, re-decoded frame 3 (since it is the LT frame for frames 8 and 9) must be kept until the ACK/NACK information arrives for frame 8 and 9. Similarly with re-decoded frame 5 which is the LT frame for 10 and 11. Re-decoded frames 3 and 7 are used to improve the estimate of frame 8. Re-decoded frame 3 and estimated frame 8 are then used to improve the estimate of frame 9. Re-decoded frame 5 and estimated frame 9 are used to improve the estimate of frame 10. Lastly, redecoded frame 5 and estimated frame 10 are used to improve the estimate of frame 11. Now the encoder can encode frame 12. So, in this example, the largest number of frames being buffered at any given time is 9 (that is, frames 1, 3, 5, 6, 7, 8, 9, 10 and 11) in addition to the frame to be encoded (frame 12).

IV. Results

We modified an existing H.263+ video codec to use a dual frame buffer with accurate half pel modeling as in [11]. The results shown were averaged over 100 random channel realizations. We studied codec behavior for varying values of delay d, packet loss probability p, and bit rate. All results are for continuous updating of the ST and LT frames. Continuous updating outperformed jump updating for most cases (but we note that continuous updating imposes somewhat heavier buffering requirements than does jump updating). In each plot, our dual frame buffer results are compared against the ROPE algorithm with feedback [6], where optimal mode switching decisions are made in a rate-distortion context, but using conventional single frame motion compensation.

Since we transmit one additional bit to the decoder to signal whether the reference buffer will be the ST or the LT frame, our coder is not standard compliant.

PSNR vs. bit rate: Fig. 4 shows results for the "Container" QCIF image sequence for continuous updating ($N = 1$ and $D = 3$) at a frame rate of 15 fps and a feedback delay of $d = 3$. Simulations show a PSNR delta in favor of dual frame prediction that ranges from 1.4dB, for a bit rate of 50kbps, to 1.3dB, for a bit rate of 150kbps. This particular sequence benefits greatly from the use of dual frame. Dual frame without ROPE estimation does poorly; it performs less well than dual frame with ROPE by 3 to 6dB.

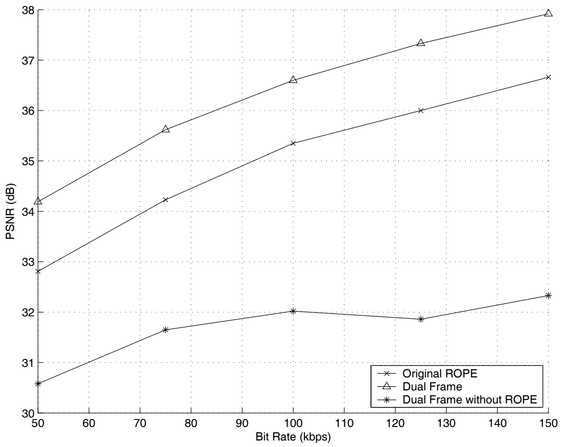

Fig. 4. PSNR Performance vs. bit rate. "Container" QCIF sequence at 15 fps, with continuous updating, a feedback delay $d = 3$ and packet loss rate $p = 10\%$.

PSNR vs. packet loss rate: Fig. 5 shows the PSNR performance for "News" QCIF for $N = 1$ with a feedback delay of $d = 6$ and a bit rate of 300kbps. The dual frame proves more robust as the error rate increases. The difference stands at 0.5dB at $p = 0.05$, and increases steadily to 1.5dB at $p = 0.25$. Without ROPE we have a loss of more than 5dB that keeps increasing as p does.

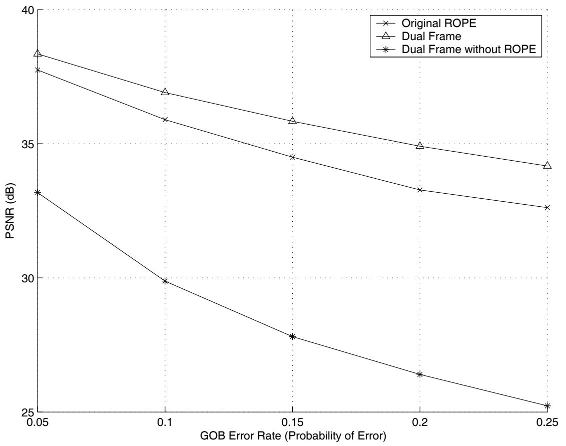

Fig. 5. PSNR Performance vs. packet loss rate. “News” QCIF sequence at 30fps, with continuous updating, a feedback delay $d = 6$ and a bit rate of 300kbps.

PSNR vs. delay: Fig. 6 examines the behavior of the image sequence “Hall” at a frame rate of 10 fps, with a packet loss rate of $p = 0.10$ and a bit rate of 90kbps. The dual frame variant shows an advantage over original ROPE that ranges from 0.65dB for small feedback delays and reaches 0.8dB for $d = 20$. The increase of d appears to favor the dual frame over single, but more experimentation with various sequences is required. Not using ROPE leads to heavy losses in PSNR that reach 6dB.

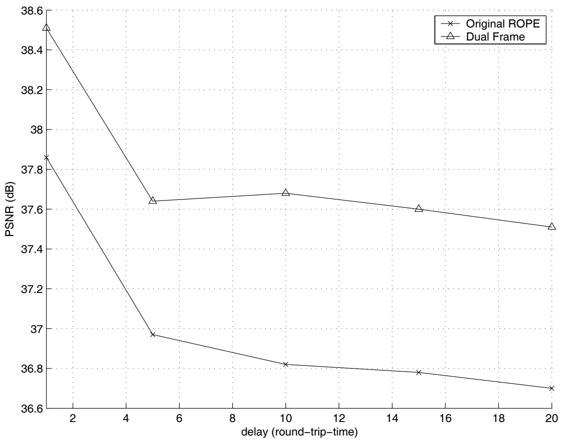

Fig. 6. PSNR Performance vs. delay. “Hall” QCIF sequence at 10 fps, with continuous updating, packet loss rate $p = 10\%$ and a bit rate of 90kbps.

V. Conclusions

The experimental results show a significant PSNR gain ranging from 0.50 to 1.6dB, for an average of about 0.90dB. The dual frame predictor withstands high packet loss rates efficiently. The experimental results showed that when feedback is employed, dual frame schemes consistently outperform single-frame ones, and the advantage tends to become more apparent as the bit rate or the packet loss rate grows large. Thus, the addition of a long-term frame buffer for motion compensation improves the encoder's compression efficiency and renders the bitstream more robust to packet drops. Using only a single extra frame buffer keeps the computational complexity relatively low.

With visual inspection of the reconstructed sequences, the dual frame predictor provides a noticeably smoother viewing experience. Background details are preserved, and packet losses generally affect only macroblocks with high motion.

Acknowledgments

We would like to thank Prof. Kenneth Rose and Dr. Rui Zhang for making available their source code for the ROPE algorithm. This work was supported in part by the National Science Foundation, the Center for Wireless Communications at UCSD, and the CoRe program of the State of California.

References

[1] M. Gothe and J. Vaisey, “Improving motion compensation using multiple temporal frames,” in *Proc. IEEE Pac. Rim Conf. on Comm., Comp. and Signal Proc.*, vol. 1, May 1993, pp. 157–160.

[2] N. Vasconcelos and A. Lippman, “Library-based image coding,” in *Proc. IEEE Int. Conf. on Ac., Speech, and Signal Proc.*, vol. v, Apr. 1994, pp. 489–492.

[3] T. Wiegand, X. Zhang, and B. Girod, “Long-term memory motion-compensated prediction,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp. 70–84, Feb. 1999.

[4] G. Côté, S. Shirani, and F. Kossentini, “Optimal mode selection and synchronization for robust video communications over error-prone networks,” *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 952–965, June 2000.

[5] D. Wu, Y. T. Hou, B. Li, W. Zhu, Y.-Q. Zhang, and H. J. Chao, “An end-to-end approach for optimal mode selection in internet video communication: Theory and application,” *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 977–995, June 2000.

[6] R. Zhang, S. L. Regunathan, and K. Rose, “Video coding with optimal inter/intra-mode switching for packet loss resilience,” *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 966–976, June 2000.

[7] T. Wiegand, N. Färber, K. Stuhlmüller, and B. Girod, “Error-resilient video transmission using long-term memory motion compensated prediction,” *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 1050–1062, June 2000.

[8] T. Stockhammer, T. Wiegand, and S. Wenger, “Optimized transmission of H.26L/JVT coded video over packet-lossy networks,” in *Proc. IEEE International Conference on Image Processing*, vol. 2, 2002, pp. 173–176.

[9] T. Fukuhara, K. Asai, and T. Murakami, “Very low bit-rate video coding with block partitioning and adaptive selection of two time-differential frame memories,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 1, pp. 212–220, Feb. 1997.

[10] M. Budagavi and J. D. Gibson, “Multiframe video coding for improved performance over wireless channels,” *IEEE Trans. Image Processing*, vol. 10, no. 2, pp. 252–265, Feb. 2001.

[11] A. Leontaris and P. C. Cosman, “Video compression with intra/inter mode switching and a dual frame buffer,” in *Proc. IEEE Data Compression Conference*, Mar. 2003, pp. 63–72.