Title

Video Compression for Lossy Packet Networks With Mode Switching and a Dual-Frame Buffer

Journal

IEEE Transactions on Image Processing, 13(7)

ISSN

1057-7149

Authors

Leontaris, A.
Cosman, P. C.

Publication Date

2004-07-01

DOI

10.1109/TIP.2004.828429

Peer reviewed

Video Compression for Lossy Packet Networks With Mode Switching and a Dual-Frame Buffer

Athanasios Leontaris, *Student Member, IEEE,* and Pamela C. Cosman, *Senior Member, IEEE*

***Abstract*—Video codecs that use motion compensation benefit greatly from the development of algorithms for near-optimal intra/inter mode switching within a rate-distortion framework. A separate development has involved the use of multiple-frame prediction, in which more than one past reference frame is available for motion estimation. In this paper, we show that using a dual-frame buffer (one short-term frame and one long-term frame available for prediction) together with intra/inter mode switching improves the compression performance of the coder. We improve the mode-switching algorithm with the use of half-pel motion vectors. In addition, we investigate the effect of feedback in making more informed and effective mode-switching decisions. Feedback information is used to limit drift errors due to packet losses by synchronizing the long-term frame buffers of both the encoder and the decoder.**

***Index Terms*—Dual-frame buffer, mode switching, multiple-frame prediction, packet-switched networks, video compression.**

I. Introduction

PACKET-SWITCHED networks have become ubiquitous and form the backbone of the Internet. These networks have been designed with delivery of data in mind [1]. Thus, protocols such as TCP provide guaranteed transmission of packets but are not well suited for real-time delivery of streaming video content [2]. UDP, on the other hand, is widely used for streaming video. Higher level protocols such as the real-time streaming protocol (RTSP) [3] were recently proposed and implemented to overcome these problems. Due to time constraints imposed by real-time operation, it is not feasible to retransmit packets which were lost due to network congestion or buffer overflows. Consequently, packet losses can severely corrupt an unprotected bitstream. The transmitted bitstream has to be organized so as to minimize corruption and error propagation due to dropped packets.

Contemporary hybrid video codecs use motion-compensated prediction to efficiently encode a raw input video stream. A block in the current frame is predicted from a displaced block in the previous frame. The difference between the original one and its prediction is compressed and transmitted along with the displacement (motion) vectors. Called *inter* coding, this is the basic approach found in the video coding standards MPEG, MPEG-2, MPEG-4 [4], H.263 [5], and the latest and state-of-the-art H.264/AVC [6].

The idea of using more than one past reference frame to improve coding efficiency is not new. An early work [7] of multiple-reference frames coding showed that the mean-squared error (MSE) between the current frame and the predicted one strictly decreases by using multiple-temporal frames for motion compensation. However, no experiments were run using a hybrid codec with quantization of the transform coefficients. Another early attempt to code an image using a so-called *library* of past frame components can be found in [8], and made use of vector quantization. Long-term memory multiple-frame prediction was again treated in [9]. In this paper, experiments were conducted on an actual hybrid video codec, and rate-distortion optimization included not only the potential motion vectors (MVs), but also the temporal delay parameter d. A window of 50 past frames was used, incurring a heavy computational penalty.

In [10], only two time-differential frames were used, thus requiring a relatively modest increase in computational complexity. We refer to this as a *dual-frame buffer*. The first buffer included the previous frame, as in many hybrid codecs, and the second one contained a reference frame from the more distant past that was periodically updated according to a predefined rule. Using this scheme in conjuction with another technique called block partitioning, they showed that it can have a positive impact on compression efficiency, despite using only one long-term frame. In [11], the authors use a linear weighted combination of two frames, primarily to enhance the error robustness of the codec. Error propagation was analyzed theoretically and it was shown that error robustness improves by using two frames. In [12], the authors use Markov chain analysis to prove that multiple frames increase error robustness. They also derive a rule to randomize the selection of the frame buffer (among a window of past ones) and, thus, inject additional error resilience into the codec.

Rate distortion-based techniques for optimal coding mode selection were studied in [13]–[16] and [17]. A novel algorithm for calculating estimated distortion due to packet losses was introduced in [18] and will be described in Section II. Robust video transmission was studied in [19]–[21]. In [21], long-term memory motion-compensated prediction was used, and distortion due to error propagation was modeled as a tree where each leaf represented different decoded versions of the same frame. The final computationally tractable model that was adopted by the authors used only three branches, reducing the accuracy of the model. Feedback performance was also investigated. In

Manuscript received March 31, 2003; revised November 24, 2003. This work was supported in part by the National Science Foundation, in part by the Center for Wireless Communications at UCSD, and in part by the CoRe program of the State of California. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Aria Nosratinia.

The authors are with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093-0407 USA (e-mail: aleontar@code.ucsd.edu; pcosman@code.ucsd.edu).

Digital Object Identifier 10.1109/TIP.2004.828429

[22], K encoder/decoder pairs were simulated under K different error patterns to model potential errors. However, even for $K = 30$ that was used, convergence was not guaranteed and the distortion estimation algorithm exhibited $O(K)$ complexity.

In this paper, we show how using a dual-frame buffer together with an algorithm for intra/inter mode switching decisions can lead to improved compression performance. We first examine performance assuming no feedback is present, and then we experiment with a more refined updating that takes into account feedback signals to effectively synchronize the long-term frame buffers of both the encoder and decoder. The paper is organized as follows. In Section II, we review the ROPE algorithm [18] for distortion estimation. In Section III, we show how this algorithm can be used in the context of a dual-frame buffer. The use of half-pel MVs is covered in Section IV. Results in the absence of feedback are presented in Section V. In Section VI, we describe the feedback extensions, with experimental results given in Section VII. Complexity is analyzed in Section VIII. Finally, conclusions are drawn in Section IX.

II. ROPE Algorithm

Recent attempts to switch coding modes according to error robustness criteria can be found in [23], [20], [22], and [18]. Our work makes use of the recursive optimal per-pixel estimate (ROPE) algorithm [18] which provides distortion estimates, which are then used for mode decision in hybrid video coders operating over packet erasure channels. In general, inter mode achieves higher compression efficiency than intra mode, at the cost of potentially severe error propagation. A single error in a past frame may corrupt all subsequent frames if inter coding is used repeatedly. This error propagation can only be stopped by transmitting and successfully receiving an intra-coded macroblock (MB). The problem that arises is how to optimally select between intra- and intercoding for each MB, such that both error resilience and coding efficiency are achieved.

We assume that the video bitstream is transmitted over a packet erasure channel (lossy packet network). Each frame is partitioned into groups of blocks (GOBs). Each GOB contains a single horizontal slice of MBs 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, but keeps 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 [24], [25]. The decoder replaces the lost MB by one from the previous frame, using as 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 three nearest MBs were all intracoded and, therefore, have no MVs), then the all-zero (0, 0) MV is used, and the lost MB is replaced with the co-located one from the previous frame.

We will now summarize the ROPE algorithm [18] in some detail as these equations will prove useful in elaborating our proposed method. Within this section, we make use of the notation and equations from [18]. 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 [18], in which it is necessary to separate out the cases of intra- and inter-coded MBs.

For an intra-coded MB, $\tilde{f}_n^i = \hat{f}_n^i$ with probability $1 - p$, corresponding to correct receipt of the packet. If the packet is lost, but the previous GOB is correct, the concealment based on the median MV leads the decoder to associate pixel i in the current frame with pixel k in the previous frame. Thus, $\tilde{f}_n^i = \tilde{f}_{n-1}^k$ with probability $p(1-p)$. Finally, if both current and previous GOB packets are lost, $\tilde{f}_n^i = \tilde{f}_{n-1}^i$ (occurs with probability p^2). So, the two moments for a pixel in an intra-coded MB are [18]

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

For an inter-coded MB, let us assume that its true MV is such that pixel i is predicted from pixel j in the previous frame. Thus, the encoder prediction of this pixel is $\hat{f}_{n-1}^j$. The prediction error e_n^i is compressed, and the quantized residue is $\hat{e}_n^i$. The encoder reconstruction is

$$\hat{f}_n^i = \hat{f}_{n-1}^j + \hat{e}_n^i. \quad (4)$$

The encoder transmits $\hat{e}_n^i$ and the MBs MV. If the packet is correctly received, the decoder knows $\hat{e}_n^i$ and the MV, but must still use its own reconstruction of pixel j in the previous frame $\tilde{f}_{n-1}^j$, which may differ from the encoder value $\hat{f}_{n-1}^j$. Thus, the decoder reconstruction of pixel i is given by

$$\tilde{f}_n^i = \tilde{f}_{n-1}^j + \hat{e}_n^i. \quad (5)$$

Again, the encoder models $\tilde{f}_{n-1}^j$ as a random variable. The derivation of the moments is similar to the intra-coded MB for the last two cases, but differs for the first case where there is no transmission error (probability $1 - p$). The first and second moments of $\tilde{f}_n^i$ for a pixel in an inter-coded MB are then given by

$$\begin{aligned} E\{\tilde{f}_n^i\} &= (1-p)\left(\hat{e}_n^i + E\{\tilde{f}_{n-1}^j\}\right) \\ &\quad + p(1-p)E\{\tilde{f}_{n-1}^k\} + p^2 E\{\tilde{f}_{n-1}^i\} \quad (6) \\ E\{(\tilde{f}_n^i)^2\} &= (1-p)\left((\hat{e}_n^i)^2 + 2\hat{e}_n^i E\{\tilde{f}_{n-1}^j\} + E\{(\tilde{f}_{n-1}^j)^2\}\right) \\ &\quad + p(1-p)E\{(\tilde{f}_{n-1}^k)^2\} + p^2 E\{(\tilde{f}_{n-1}^i)^2\}. \quad (7) \end{aligned}$$

These recursions are performed at the *encoder* in order to calculate the expected distortion at the *decoder*. The encoder can exploit this result in its encoding decisions, to optimally choose the coding mode for each MB. The expectation for each pixel is calculated as a weighted sum (due to the probabilities) of pixel expectations from the previous frame, prediction residuals, and intra-coefficients.

A. Rate-Distortion Framework

The ROPE algorithm estimates the expected distortion, due to both compression and transmission errors, to be used for optimal mode switching. The encoder switches between intra- or intercoding on a MB basis, in an optimal fashion for a given bit rate and packet loss rate. The goal is to minimize the total distortion D subject to a bit-rate constraint R. Using a Lagrange multiplier λ, the ROPE algorithm minimizes the total cost $J = D + \lambda R$. Individual MB contributions to this cost are additive, thus, it can be minimized on a MB basis. Therefore, the encoding mode for each MB is chosen by minimizing

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

where the distortion D_{MB} of the MB is the sum of the distortion contributions of the individual pixels. Rate control is achieved by modifying λ as in [26]. Both the *coding mode* and the *quantization step size QP* are chosen to minimize the Lagrangian cost. This is computationally complex for the encoder, but it enhances coding efficiency. The resulting bitstream is compatible with a standard compliant decoder.

We note that while the ROPE algorithm is optimal under the given assumptions, there is potential for improvement by incorporating the MV choice into the rate-distortion framework, or by correctly estimating distortion for half-pel vectors (the algorithm only models distortion for integer MVs).

III. Dual-Frame Buffer Extension

Our research has focused on using a dual-frame buffer together with optimal mode switching within a rate-distortion framework. The basic use of the dual-frame buffer is as follows. While encoding frame n, the encoder and decoder both maintain two reference frames in memory. The short-term reference frame is frame $n-1$. The long-term reference frame is, say, frame $n-k$, where k may be variable, but is always greater than 1. Each MB can be encoded in one of three coding modes: intra-coding, inter-coding using the short-term buffer (inter-ST-coding), and inter-coding using the long-term buffer (inter-LT-coding). This is illustrated in Fig. 1. The choice among these three will be made using an extended version of the ROPE algorithm, as described below. Once the coding mode is chosen, the syntax for encoding the bit stream is almost identical to the standard case of the single-frame (SF) 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.

The choice among the three coding modes does not, of course, need to be done using an extension of the ROPE algorithm. A naive approach would be to use a traditional distortion estimator that evaluates the distortion from motion compensation and quantization alone. However, experimental results showed a substantial advantage (up to 3–4 dB) to using a rate-distortion-based decision with a ROPE distortion estimator instead of a rate distortion-based decision with a traditional distortion estimator. This was true for both SF and dual-frame coders. Given the substantial benefit to using the ROPE distortion estimator over a traditional distortion estimator, this paper focuses on extending the ROPE algorithm to work with a dual-frame coder, comparing it against a SF ROPE coder.

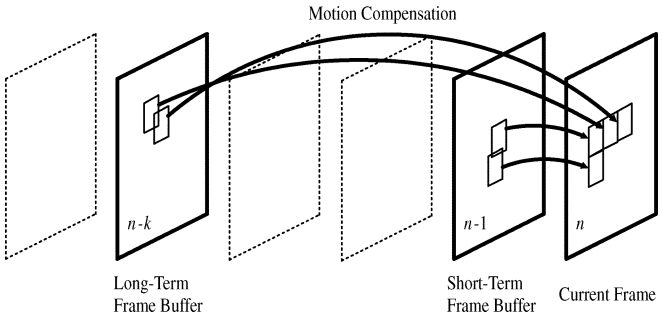

Fig. 1. Dual-frame buffer motion compensation.

We now describe how the long-term reference frame is chosen. In one approach, which we call *jump updating*, the long-term reference frame varies from as recent as frame $n-2$ to as old as frame $n-N-1$. When encoding frame n, if the long-term 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 long-term reference frame will jump forward by N to frame $n-1$. The long-term 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 [10].

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. These two approaches are depicted in Fig. 2.

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. Clearly, the most general updating strategy would have no fixed N or D; rather, 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.

Let us now elaborate on how the choice is made among the coding modes. As before, we use f_n, $\hat{f}_n$, and $\tilde{f}_n$ to denote the

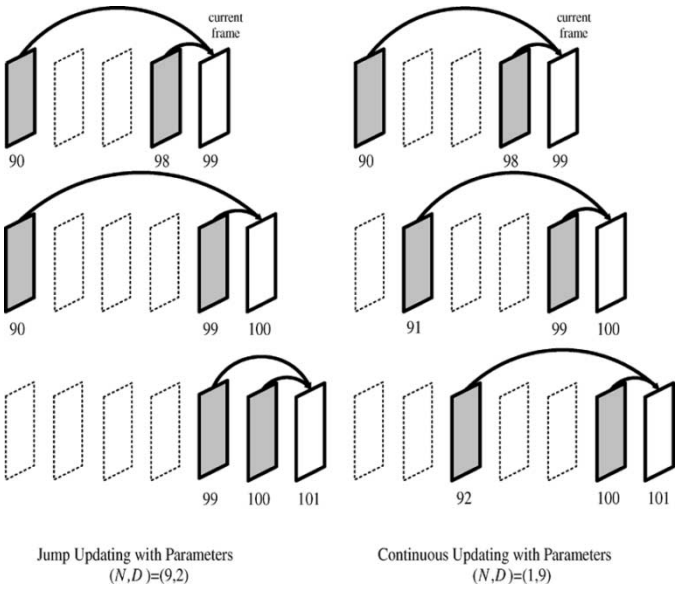

Fig. 2. Two different dual-frame buffer approaches. In the top row, frame 99 is being predicted from frames 98 and 90. In the middle row, the current frame to be encoded is frame 100. With the jump updating approach, frames 99 and 90 are used for prediction. With the continuous updating approach, frames 99 and 91 are used. However, as we examine the bottom row, we observe that jump updating takes place when 101 is encoded. Thus, the new long-term frame buffer will be frame 99, while for the continuous updating approach, we will use 92.

original frame n, the encoder reconstruction of the compressed frame, and the decoder version of the frame, respectively. We assume that the long-term frame buffer was updated l frames ago. Thus, it contains $\hat{f}_{n-l}$ at the transmitter and $\tilde{f}_{n-l}$ at the receiver. The expected distortion for pixel i in frame n is given by (1).

To compute the moments in (1), the recursion steps for pixels in intra-coded and inter-ST-coded MBs are identical to the corresponding steps in the original ROPE algorithm. For a pixel in an inter-LT-coded MB, we assume that the true MV of the MB is such that pixel i in frame n is predicted from pixel j in frame $n-l$, where $l>1$. The encoder prediction of this pixel is $\hat{f}^j_{n-l}$. The prediction error e^i_n is compressed, and the quantized residue is denoted by $\hat{e}^i_n$. The encoder reconstruction of the pixel is

$$\hat{f}^i_n = \hat{e}^i_n + \hat{f}^j_{n-l}. \tag{9}$$

As the receiver does not have access to $\hat{f}^j_{n-l}$, it uses $\tilde{f}^j_{n-l}$

$$\tilde{f}^i_n = \hat{e}^i_n + \tilde{f}^j_{n-l}. \tag{10}$$

When the MB is lost, the median MV from the three nearest MBs is calculated and used to associate pixel i in the current frame with pixel k in the previous frame. Using the same arguments as in the original ROPE algorithm, we compute the first and second moments of $\tilde{f}^i_n$ for a pixel in an inter-LT-coded MB

$$\begin{aligned} E\{\tilde{f}^i_n\} &= (1-p)\left(\hat{e}^i_n + E\{\tilde{f}^j_{n-l}\}\right) \\ &\quad + p(1-p)E\{\tilde{f}^k_{n-1}\} + p^2 E\{\tilde{f}^i_{n-1}\} \qquad (11) \\ E\{(\tilde{f}^i_n)^2\} &= (1-p)\left((\hat{e}^i_n)^2 + 2\hat{e}^i_n E\{\tilde{f}^j_{n-l}\} + E\{(\tilde{f}^j_{n-l})^2\}\right) \\ &\quad + p(1-p)E\{(\tilde{f}^k_{n-1})^2\} + p^2 E\{(\tilde{f}^i_{n-1})^2\}. \qquad (12) \end{aligned}$$

We note that error concealment is still done using the *previous* frame $n-1$ and not the long-term frame. This is done regardless of whether the three MBs above are inter-ST-coded or inter-LT-coded, or some combination of the two. The MVs may be uncorrelated. If the upper GOB is also lost, we conceal the MB using the co-located block from the previous frame.

Using an additional reference frame (LT) has some drawbacks with respect to MV compression efficiency when compared to SF. There is a bit-rate loss due to inaccurate prediction of MVs from the neighboring and potentially uncorrelated MVs. By neighboring MV, we mean the MV of the MB on the left of the one being coded. During coding, we do not predict MVs using the MVs above because we wish the GOBs to be decodable independently of each other. The first MB of each GOB uses no prediction for the MVs. For those MBs where the MV points to the same reference frame as the neighbor (and only for those MBs), we obtain an MV coding efficiency equal to that of SF approaches. As an alternative approach, we tried predicting the MV using the neighbor only when the neighbor corresponded to the same reference frame. When the neighbor did not use the same reference, the MV would be coded without prediction. Experimentally, this did not do as well. The explanation for this is that with relatively small values for N, MVs pointing to either the short-term or the long-term frame buffer tend to have similar values, so it is better to use them for prediction than to code MVs without prediction. However, they are not as similar as are MVs in SF motion compensation, so there is still a loss in MV compression efficiency. As will be seen in the results section, this loss in MV compression efficiency is more than made up for in other ways by the dual-frame coder.

Compression efficiency will also suffer due to the need to transmit one bit for every inter-coded MB to specify the frame buffer (this overhead could be reduced by using run length coding on the bits, but we do not do this as it incurs penalties in terms of buffering at the decoder and a risk of catastrophic error if the RLC-encoded frame buffer selection stream is lost). Nonetheless, as experimental results will show, the rate-distortion optimization models these additional bits, and is still able to yield superior compression performance.

The requirement to encode and decode this additional bit (for selecting between ST/LT), clearly makes this proposed scheme *not* a H.263+ compliant codec. Since H.264 already supports multiple-frame prediction, there is no compliance problem. However, a straightforward application of ROPE on H.264 without any modifications is not wise. Apart from the half-, quarter-, and eighth-pel accuracy present within H.264, which would have to be modeled (see Section IV), there is also the problem of the loop filter, and additional concealment modes, which would require evaluating multiple-product expectations (correlations).

Since the quantization parameter *QP* takes values from 1 to 31, the coder optimizes over 62 potential combinations of coding modes (intra or inter) and quantization parameters by calculating the estimated distortion using ROPE. With the extra coding mode inter-LT, the search for optimal coding parameters is conducted over 93 combinations. There is a computational increase of approximately 50% for the rate-distortion optimization portion of the encoder. Furthermore, motion estimation

complexity is approximately doubled. Hence, the total encoding time of the modified encoder is roughly 1.8 times that of the baseline ROPE encoder. Further analysis on computational complexity is provided in Section VIII.

IV. Half-Pixel Approximation Extension

The use of integer MVs limits the reference choices in the previous frame. Most video codecs show a performance advantage when half-pel MVs are implemented, as the encoder is now presented with many more options in the search for the best-match block. The use of an additional reference frame likewise presents the encoder with more options for the best match block. We wished to see how the gains from an additional frame buffer compared to those from adding a half-pel grid, and also whether the two approaches could be used together for greater benefit.

The use of a half-pel grid in a standard video codec requires the generation of the half-pel values using some kind of interpolation, and then requires a four-fold increase in the MV search. However, simply adding a half-pel grid within the ROPE algorithm, and attempting to run the optimal mode switching over it, incurs a far more substantial complexity penalty than this, as discussed below.

Since the accurate use of a half-pel grid is prohibitive, another approach would be to use a half-pel grid only for finding and transmitting MVs, but to leave it out of the ROPE distortion calculation altogether. This is what is done in [18], which we call the unmodeled half-pel, and it provides some improvement over the use of strictly integer MVs. However, as we will now discuss, an approximate modeling of the half-pels within the ROPE algorithm provides further improvement, while avoiding the computational complexity of the fully accurate modeling of a half-pel grid in ROPE.

We assume that error concealment is still done using only the integer portion of the MVs, and therefore (2) and (3) for the intra-coded MBs are unchanged. Returning to (6) and (7) for the inter-coded MBs, we see the terms $\hat{e}_n^i$, $E\{\tilde{f}_{n-1}^k\}$, $E\{\tilde{f}_{n-1}^i\}$, $E\{(\tilde{f}_{n-1}^k)^2\}$ and $E\{(\tilde{f}_{n-1}^i)^2\}$ remain unchanged. However, the calculation of $E\{\tilde{f}_{n-1}^j\}$ and $E\{(\tilde{f}_{n-1}^j)^2\}$ has become critical. Pixel coordinate j now points to a position in an interpolated grid that covers an area four times that of the original image.

For this calculation, we differentiate among three types of pixels on the half-pel grid: pixels that coincide with actual (original) pixel positions (called integer-indexed pixels, they do not need to be interpolated), pixels that lie between two integer-indexed pixels (either horizontally or vertically), and pixels that lie diagonally between four integer-indexed pixels. We use bilinear interpolation, so the interpolated value is simply the average of the two or four neighboring integer-indexed pixels.

For the integer-indexed pixels, the recursion equations are identical to those of the baseline ROPE algorithm, and the estimation is optimal.

A. Horizontally or Vertically Interpolated Pixel

For a horizontally or vertically interpolated pixel, we assume that j on the interpolated pixel domain corresponds to a pixel that was interpolated using pixels k_1 and k_2 in the original pixel domain. We define the following abbreviations. Let $\mu_i = E\{\tilde{f}_{n-1}^{k_i}\}$ denote the estimate (mean-value) of the pixel with coordinates k_i in frame $n-1$, $\mu_{i,j} = E\{\tilde{f}_{n-1}^{k_i}\tilde{f}_{n-1}^{k_j}\}$ denote the correlation (expectation of product) between pixels k_i and k_j, and $\sigma_i = E\{(\tilde{f}_{n-1}^{k_i})^2\}$ denote the mean-squared value of pixel k_i. The first moment is computationally tractable

$$E\{\tilde{f}_{n-1}^j\} = \frac{1}{2}\left[1 + \mu_1 + \mu_2\right]. \tag{13}$$

However, the expression for the second moment is

$$E\{(\tilde{f}_{n-1}^j)^2\} = \frac{1}{4}\left[1 + \sigma_1 + \sigma_2 + 2\mu_1 + 2\mu_2 + 2\mu_{1,2}\right]. \tag{14}$$

The last term requires calculating the correlation of matrices whose horizontal/vertical dimension equals the number of pixels in the image. This is computationally infeasible for images of typical size. The second moment can be bounded using the cosine (Cauchy–Schwartz) inequality

$$E\{(\tilde{f}_{n-1}^j)^2\} \leq \frac{1}{4}\left[1 + \sigma_1 + \sigma_2 + 2\mu_1 + 2\mu_2 + 2\sqrt{\sigma_1\sigma_2}\right] \tag{15}$$

and we will approximate it by setting the inequality to be an equality. This worked well, perhaps because the (image domain) pixel values are always positive, and so correlations tend to be close to the upper bound, which was also verified by our experimental results. During our simulations, we also experimented with multiplying the Cauchy–Schwartz-derived upper bound with various constants $c < 1$, such as $c = 0.50$; however, this did not always perform as well as the upper bound.

B. Diagonally Interpolated Pixel

For a diagonally interpolated pixel, we assume that j on the interpolated pixel grid is the result of interpolating pixels k_1, k_2, k_3, and k_4 in the original pixel domain. The first moment can be computed exactly as

$$E\{\tilde{f}_{n-1}^j\} = \frac{1}{4}\left[2 + \mu_1 + \mu_2 + \mu_3 + \mu_4\right]. \tag{16}$$

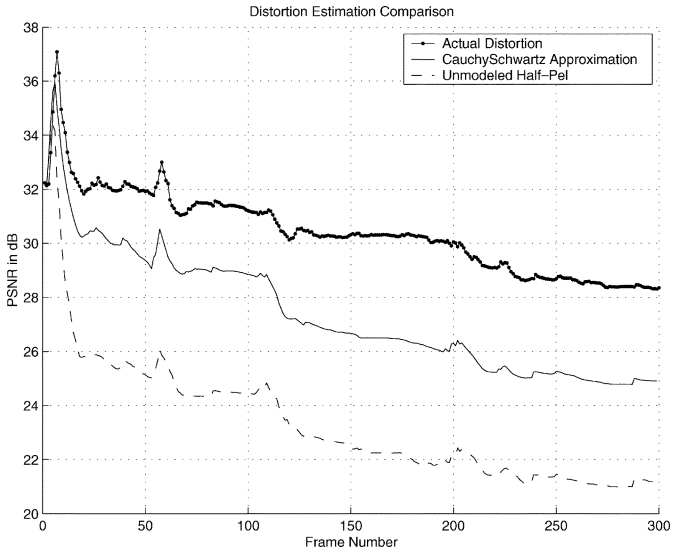

Fig. 3. Distortion estimation comparison.

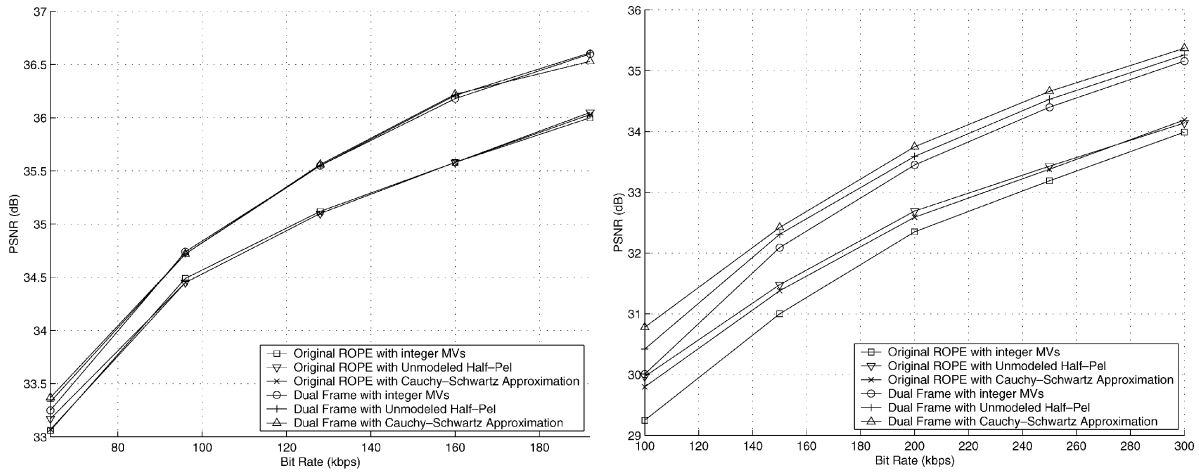

Fig. 4. PSNR performance versus bit rate. (a) "Hall" QCIF sequence at 15 fps, with continuous update parameter $D = 3$ and packet loss rate $p = 20\%$. (b) "News" QCIF at 30 fps with continuous update parameter $D = 5$ and packet loss rate $p = 15\%$.

The accurate, but intractable, expression for the second moment is

$$E\{(\tilde{f}_{n-1}^{j})^2\} = \frac{1}{16}\left[4 + \sum_{i=1}^{4}(\sigma_i + 4\mu_i) + 2\Big(\mu_{1,2} + \mu_{1,3} + \mu_{1,4} + \mu_{2,3} + \mu_{2,4} + \mu_{3,4}\Big)\right]. \quad (17)$$

Applying the same approximation as with the horizontal/vertical case, we obtain

$$E\{(\tilde{f}_{n-1}^{j})^2\} \le \frac{1}{16}\left[4 + \sum_{i=1}^{4}(\sigma_i + 4\mu_i) + 2\sqrt{\sigma_1\sigma_2} + 2\sqrt{\sigma_1\sigma_3} + 2\sqrt{\sigma_1\sigma_4} + 2\sqrt{\sigma_2\sigma_3} + 2\sqrt{\sigma_2\sigma_4} + 2\sqrt{\sigma_3\sigma_4}\right] \quad (18)$$

and, again, we use this upper limit to approximate the second moment. In what follows, we refer to this as the Cauchy–Schwartz approximation.

C. Distortion Estimation

We investigated the accuracy of our distortion approximation for half-pel MVs. The enhanced accuracy provided when the Cauchy–Schwartz inequality is employed, is depicted in Fig. 3. To obtain this graph we constrained the mode decisions to use distortion only due to quantization. No estimated distortion was used so as to make the encoder independent of the accuracy of either method. The encoder optimized its stream only with regard to compression efficiency, employing half-pel MVs and applying errors with $p = 10\%$. Concurrently, the original ROPE algorithm and the modified one with the Cauchy–Schwartz approximation estimated the resulting distortion. Our modification enables a more accurate estimate.

For integer MVs, the distortion estimation of the classical ROPE algorithm is very accurate, within 0.1–0.2 dB of the actual distortion. In Fig. 3, where half-pel vectors are applied, such an accuracy can no longer be obtained. Nevertheless, the gain in estimation accuracy by using the Cauchy–Schwartz Approximation instead of the Unmodeled Half-Pel is quite noticeable.

V. Results in the Absence of Feedback

We modified an existing H.263+ video codec [5], [27] in two ways. In the case of SF motion compensation, we used the ROPE algorithm to estimate distortion for mode-switching decisions. The resulting bitstream is fully compliant with the H.263+ standard. Second, we modified the H.263+ codec to make use of one additional (long-term) frame buffer. For both the SF and dual-frame cases, we measured the performance for integer and half-pel MVs. The half-pel results are of two types: one where the half-pel vectors are used but are not modeled in the recursive error equations, and the other where the half-pel vectors are used and are modeled using the approximations given above. We refer to these as the unmodeled half-pel and Cauchy–Schwartz approximation.

We use N to denote the jump update parameter, and D to denote the temporal distance of the long-term frame buffer in the continuous updating case. N and D were kept small to increase MV correlation and, thus, improve MV coding efficiency. The GOB packet error probability was tested with values of $p = 0.05$, 0.10, 0.15, 0.20, and 0.25. The resulting dual-frame encoder is not standard compliant [27], as it must send an additional bit for every inter-coded MB to signal the use of the short-term or long-term frame buffer. The test sequences used are standard QCIF (176×144) image sequences at frame rates of 10, 15, and 30 fps. The results shown have been averaged using 100 random channel realizations (error patterns) to achieve performance consistency. The same error patterns were used for all codec variants.

A. PSNR versus Bit Rate

In Fig. 4(a) we examine the performance of the variants for "Hall." This particular sequence is rather static and does not benefit from the use of half pel MVs (the percentage of nonzero MVs per frame is less than 4%). The gains of dual-frame increase with bit rate and quickly reach 0.6 dB. A different situation is depicted in Fig. 4(b) for "News," where even the lowest performing dual-frame version easily provides higher PSNR

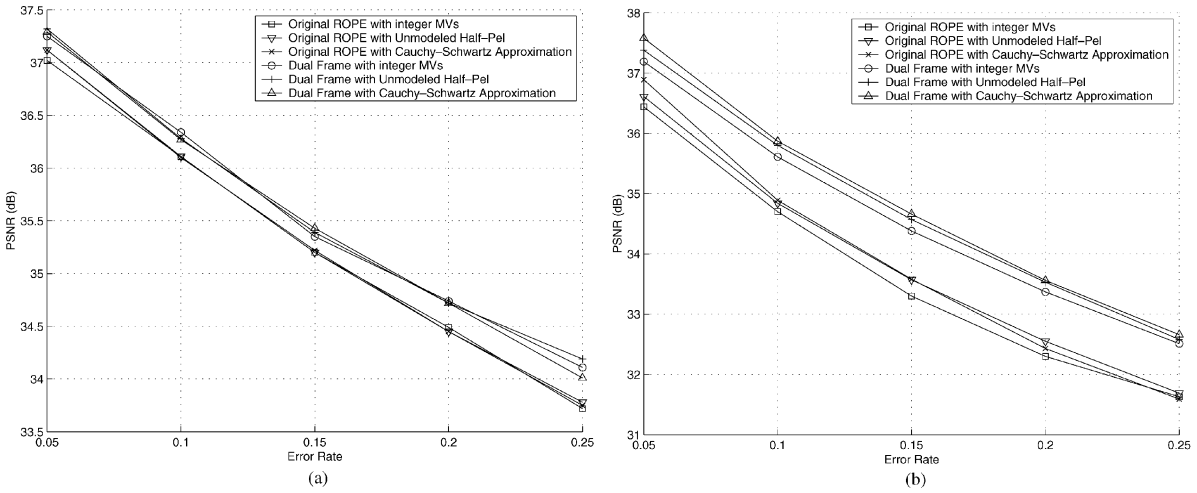

Fig. 5. PSNR performance versus error rate. (a) "Hall" QCIF sequence at 15 fps, with continuous update parameter $D = 3$ and bit-rate 96 kbps. (b) "News" QCIF at 15 fps with continuous update parameter $D = 3$ and a bit rate of 200 kbps.

than any SF approach does. Gains begin at 0.8 dB for low rates and quickly reach 1.2 dB.

Simulations using "Carphone" and "Container" yielded an average improvement of 0.4 and 0.6 dB, respectively.

B. PSNR versus Packet Loss Rate

Fig. 5(a) depicts the performance for "Hall" QCIF at 15 fps. As we pointed out for Fig. 4(a), there is no gain by using half pels. Dual frames outperform, for these particular parameters, SFs by up to 0.5 dB. The gain increases slightly with p. Similarly, in Fig. 5(b), we can observe how packet losses affect performance for the "News" image sequence at 15 fps. For both single and dual-frame methods, Cauchy–Schwartz provides a slight advantage. The performance gap between single and dual-frame approaches is approximately 0.8 dB for $p = 0.05$ and reaches 1 dB as the error rate increases.

Gains of 0.4–0.5 dB were similarly obtained for "Carphone" and "Silent." We also observed that errors are far more destructive in a lower frame-rate case than in a higher frame-rate one. When adjacent frames are more distant temporally, they are less correlated, and the respective MVs have generally higher and more varying values and are, thus, more difficult to predict. Hence, error concealment that uses estimated or all-zero MVs does much worse compared to the full frame-rate case. Some additional results can be found in [28].

C. MV Optimization

For comparison purposes, we provide some experimental results where the selection of the MVs was also incorporated within the R-D mode decision, at an enormous computational cost. The search for optimal coding parameters is conducted over 89 373 combinations (31 quantization parameter values, three coding modes, and $31 \times 31 = 961$ possible MVs), rather than just over 93. The results are, however, a good indication of the optimal attainable performance. Indicative experimental results can be seen in Fig. 6, where only quantization distortion was employed, and not the one estimated by ROPE. We can comment that even for high motion sequences such as "Foreman," the gain of 0.35 dB is definitely not worth the enormous increase in computational complexity. One of the reasons for the small gain is that MSE is often not a reliable measure of block similarity compared to SAD and, second, that motion estimation does sufficiently well at finding near-optimal MVs, so that exhaustive RD search will not yield much.

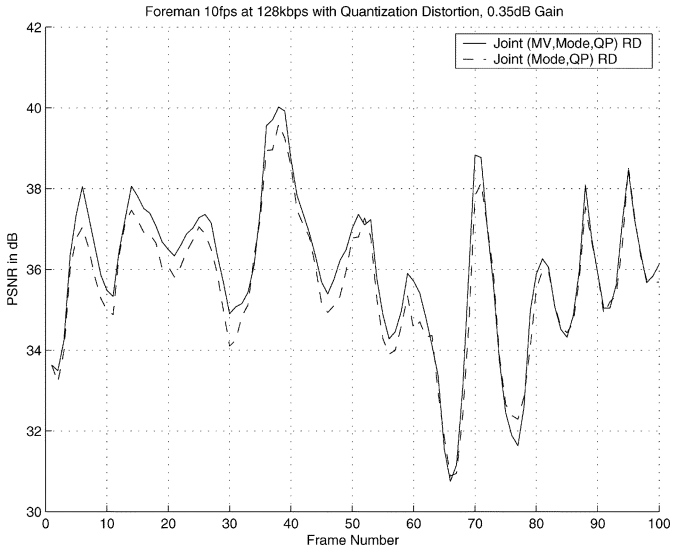

Fig. 6. Rate-distortion optimization of MV selection.

VI. Feedback Extensions

Experimental results in [18] showed that the intelligent use of feedback information (acknowledgment of received packets) can lead to substantial improvements in performance. The ROPE algorithm estimates reconstructed pixel values that incorporate potential error propagation due to packet losses. The estimates of pixel values are made by using (2), (6), and (11) for intra, inter-ST, and inter-LT coded blocks respectively. These estimates are initialized at the begining of the video sequence by assuming that the first frame is always received unharmed. 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 will use the

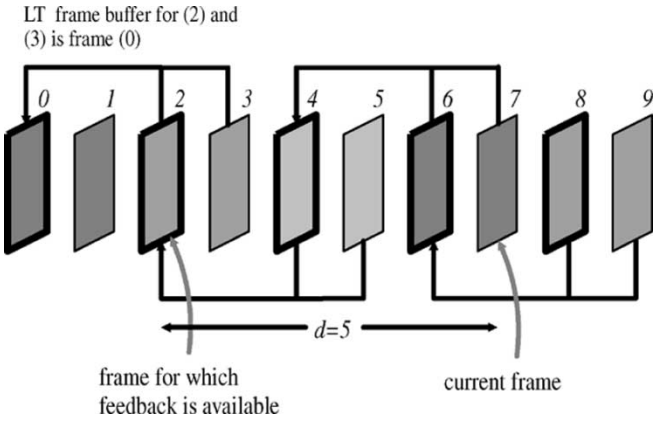

Fig. 7. Example of approach A, where $N = 2$ and $d = 5$.

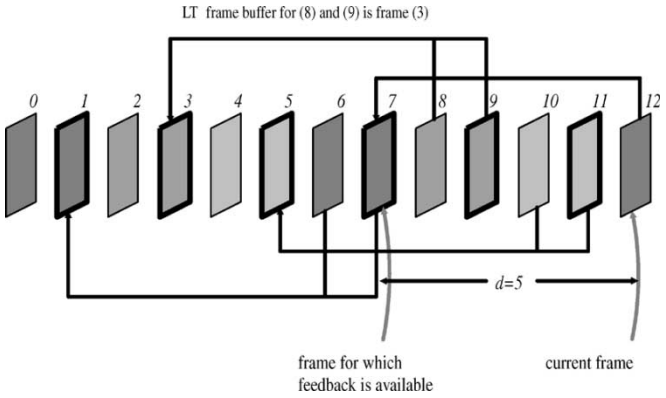

Fig. 8. Example of approach B, where $N = 2$ and $d = 5$.

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. A "re-decoded" frame is one at the encoder that is identical to the decoder version, whereas 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. With feedback information, estimates of pixel values in intermediate frames are still made using (2), (6), and (11) for intra-, inter-ST, and inter-LT coded MBs as before; however, now the information about past decoder frames required by these equations can be reinitialized using the ACKed/NACKed re-decoded frames. Then, the encoder can recalculate the pixel estimates much more reliably and track potential errors for the last d frames. The actual prediction residuals or intra coefficients are fed into the ROPE estimation algorithm where the reference frames are either ROPE estimates that also were calculated recursively, or re-decoded frames. This approach was applied to a traditional SF reference video coder in [18] with positive results. However, it lends itself to considerable improvement through the use of a dual-frame buffer.

An example is illustrated in Fig. 7. Here, the jump update parameter and the feedback delay are respectively $N = 2$ and $d = 5$. The jump update parameter $N = 2$ means that frame 0 will be the long-term reference for frames 2 and 3, frame 2 will be the long-term reference for frames 4 and 5, and frame 4 will be used for frames 6 and 7. Frames that serve as long-term frame buffers for future frames are highlighted with a thicker black outline.

Since $d = 5$, at the start of encoding frame 7, frame 2 will be re-decoded, and this newly re-decoded frame can be promptly used to update the estimates of frames 3, 4, 5, and 6. For encoding frame 7, the long-term frame is frame 4, and the short-term one is frame 6, and the new estimates of these two frames will be used by the encoder to calculate the expected distortion due to packet drops for frame 7. This jump updating, which we call approach A, outperforms both the SF feedback variants, and the dual-frame case without feedback, as the feedback allows us to improve the estimates of the ST and LT frames.

An alternative approach is to make the long-term frame buffer move forward to contain the closest *exactly known* frame, that is the $(i-d)$ frame. The feedback here allows us to improve the estimate of the ST frame, and reduce the estimation error for the LT frame to zero. We ensure that *both* the encoder and decoder long-term frame buffers always contain an *identical* reconstruction. With a delay of d, we can use either a general (N, D) updating strategy with $D = d$ and $N > 1$ (approach B), or a continuous updating strategy with $D = d$ and $N = 1$ (approach C). An example of approach B for $N = 2$ and $d = 5$ is depicted in Fig. 8. In Fig. 8, frame 12 is currently being encoded. Its LT frame is frame 7 which has also 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, the re-decoded 7 and the re-decoded 3 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 can totally eliminate drift errors caused by packet drop accumulation. Inter-LT-encoded MBs, if they arrive, will be reconstructed in an identical manner at the encoder and decoder. Normally, this is only guaranteed by transmitting intra-coded MBs. 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 the major difference from the original ROPE plus feedback case. 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, and use a more realistic reference frame. As we will see, the codec performs very well under a variety of conditions.

VII. Feedback Results

As before, 100 random channel realizations were run. In addition to examining performance as a function of bit rate and of packet loss rate, we now wish to study the behavior of the codec for varying values of delay d.

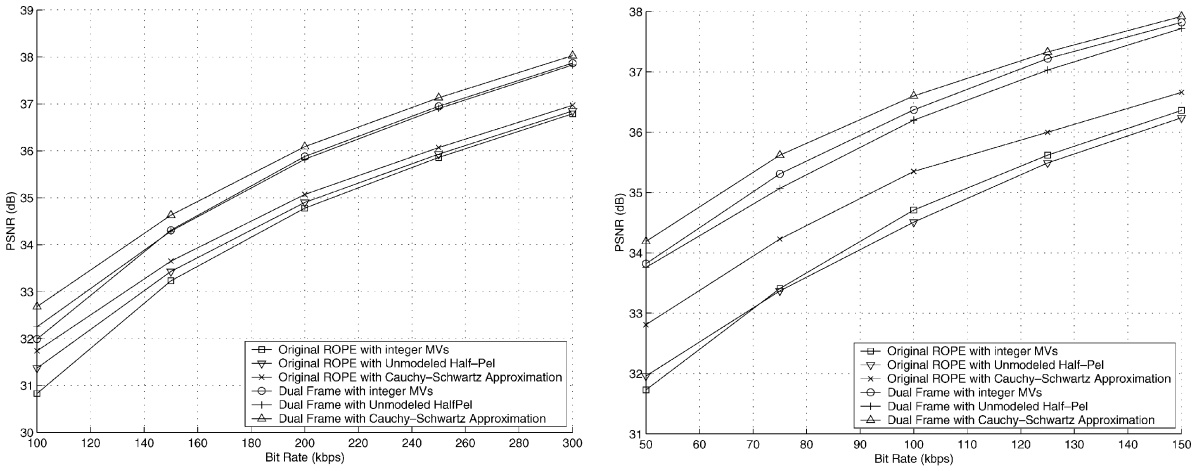

Fig. 9. PSNR performance versus bit rate. (a) "News" QCIF sequence at 30 fps, with continuous updating, a feedback delay $d = 5$ and packet loss rate $p = 10\%$. (b) "Container" QCIF at 15 fps with continuous updating, a feedback delay $d = 3$ and packet loss rate $p = 10\%$.

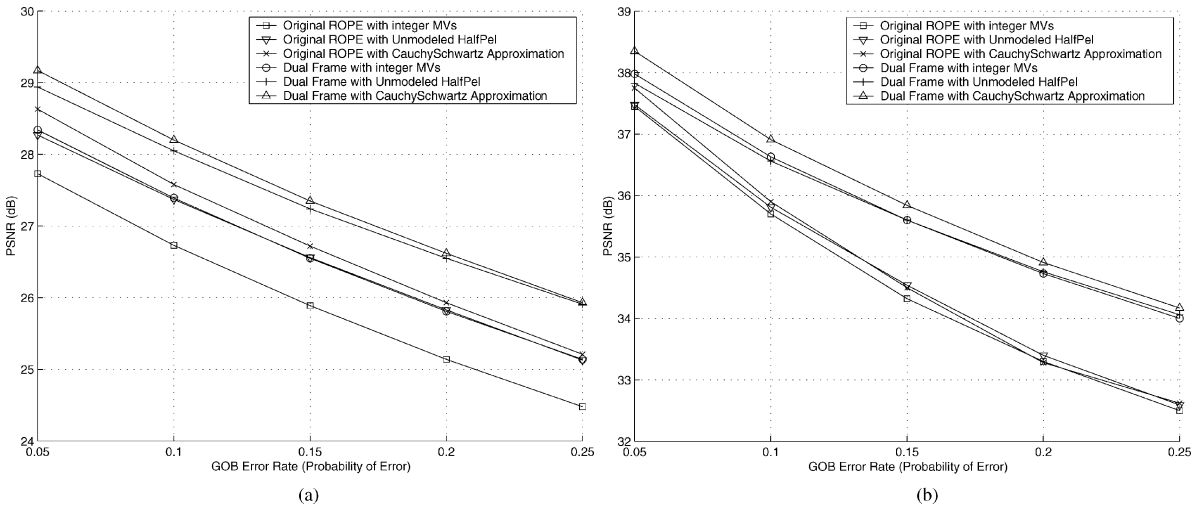

Fig. 10. PSNR performance versus packet loss rate. (a) "Foreman" QCIF sequence at 30 fps, with continuous updating, a feedback delay $d = 3$ and a bit rate of 100 kbps. (b) "News" QCIF at 30 fps with continuous updating, a feedback delay $d = 6$ and a bit rate of 300 kbps.

We note that continuous updating outperformed jump updating for four out of six image sequences. In particular, both approaches B and C outperformed approach A. However, all of them consistently outperformed SF ROPE with feedback.

A. PSNR versus Bit Rate

Fig. 9(a) shows results for the "News" sequence at 30 fps with delay parameter $d = 5$ and continuous updating (approach C). For low rates in particular, performance is significantly enhanced through the use of half-pels. With the Cauchy–Schwartz approximation, the PSNR improvement between single and dual frames grows from 0.9 dB to more than 1.1 dB as the bit rate increases. The dual-frame approach exhibits consistent performance gains as more rate is allocated. The SF variants, however, do not yield comparable gains.

In Fig. 9(b), "Container" QCIF at 15 fps and $d = 3$ was examined with approach C. This sequence benefits considerably from the Cauchy–Schwartz approximation. In the SF case, the performance advantage is almost equal to 1 dB. Simulations show a PSNR gain of almost 1.2 dB in favor of the dual-frame scheme.

Experimental results for the "Carphone" sequence showed a 0.5 dB advantage, while "Silent" QCIF showed a performance difference of 0.6 dB for high bit rates.

B. PSNR versus Packet Loss Rate

Fig. 10(a) shows the PSNR performance for "Foreman" QCIF for $N = 1$ and delay $d = 3$ for a bit rate of 100 kbps (approach C). SF ROPE in conjunction with the unmodeled half-pel outperforms dual frame with integer MVs. However, when half-pel vectors are applied to dual frame, our coder outperforms the original by up to 0.7 dB. Every dual-frame variant outperforms the corresponding SF one. The difference between the Cauchy–Schwartz versions stands at 0.5 dB at $p = 5\%$ and reaches more than 0.7 dB at $p = 25\%$.

In Fig. 10(b) ("News" image sequence at 30 fps, d = 6,300 kbps, approach C) we see that half-pixel motion estimation provides negligible to no gain against the integer version. While the performance improvement stands at only 0.6 dB at $p = 0.05$, it increases as the error rate does and reaches 1.1 dB at $p = 0.10$, 1.3 dB at $p = 0.15$, and roughly

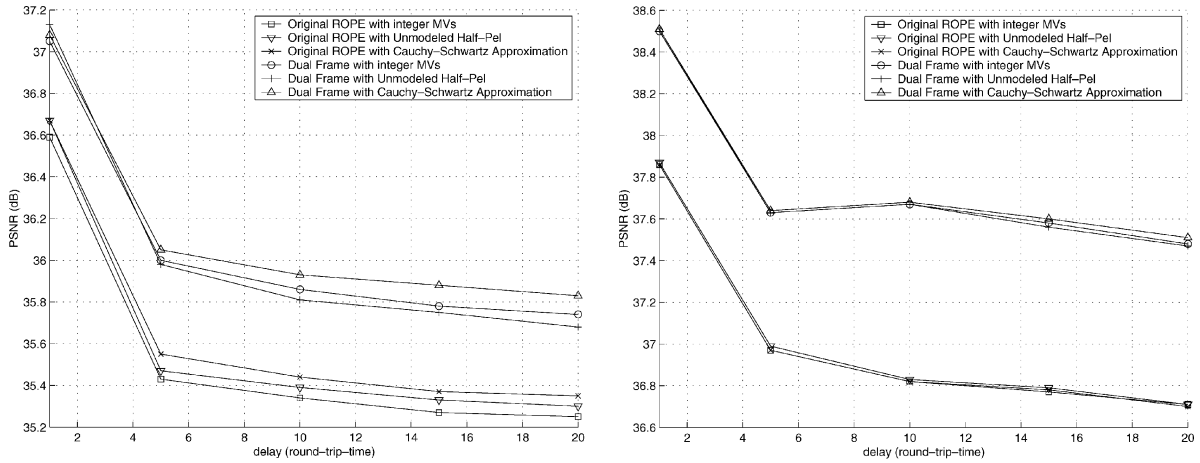

Fig. 11. PSNR performance versus delay. (a) "Silent" QCIF sequence at 10 fps, with continuous updating, packet loss rate $p = 10\%$ and a bit rate of 150 kbps. (b) "Hall" QCIF at 10 fps, with continuous updating, packet loss rate $p = 10\%$ and a bit rate of 90 kbps.

1.6 dB at $p = 0.20$ and $p = 0.25$, which means the dual frame renders the bitstream more robust to severe packet losses.

Additional simulations on the image sequences "Carphone" and "Silent" at 15 and 10 fps, respectively, yielded gains of approximately 0.5 dB.

C. PSNR versus Delay

Fig. 11(a) examines the behavior of the image sequence "Silent" at a frame rate of 10 fps, an error probability $p = 0.10$, a bit rate of 150 kbps, and using approach C. Performance is constant for $d > 10$. The positive effect of the Cauchy–Schwartz approximation reaches 0.15 dB for dual frames. The dual-frame variants show an advantage of more than 0.5 dB over SF variants.

Performance results for "Hall" at 10 fps, approach C, are depicted in Fig. 11(b). The Cauchy–Schwartz approximation fails to provide any gain in the dual-frame case, which is attributed to the fact that "Hall" is a relatively static video sequence with limited motion. However, we see that dual frame outperforms SF versions by a margin that ranges from 0.65 to more than 0.8 dB.

At the same time, "Carphone" produced gains of 0.45 dB, while "News" yielded an improvement ranging from 1.4–2 dB in favor of the dual-frame variants.

D. Dual-Frame and Half-Pel Comparison

The experimental results show that in the majority of cases, the performance gains from using Cauchy–Schwartz for ROPE estimation and from using dual-reference frames are approximately additive. However, the gain by only using dual frame is much more substantial than that of using only Cauchy–Schwartz. If just one of them would be implementable, then it is a question of computational and memory constraints, where the complexity analysis shows clearly that a dual-frame reference is costlier.

VIII. Complexity Analysis

Motion estimation (ME) is a major bottleneck in any video coder design, and implementation of real-time video codecs, especially for wireless and power-limited devices, places a limit on the use of computational resources. However, it is not the sole one. We now analyze the computational and memory requirements of our proposed scheme. In this section, we consider multiple-frame prediction with n reference frames, so we can compare with our dual-frame scheme ($n = 2$).

A. Computational Requirements

No Feedback: The *encoder* performs ME, motion compensation (MC), ROPE estimation, rate-distortion optimization, MV coding, and coefficient quantization (CQ) and de-quantization (DQ). As we will see, some of those parts invoke other functions within their execution.

The ME segment entails searching for the best match 16×16 MB over a range of $[-15, 15]$. Hence, the *optimal* integer MV is obtained. MV selection is further refined by searching over a range of $[-1, 1]$ for the best half-pel refinement vector. Let us denote its computational complexity as C_{ME} for each MB. *All* complexities C presented in this analysis are *per MB*. The MC segment reconstructs a MB at a computational cost of C_{MC}. It is obvious that $C_{\mathrm{MC}} \ll C_{\mathrm{ME}}$.

In the ROPE estimation segment, we have to differentiate among two cases: intra and inter. Intra-MBs are relatively easier to estimate at a computational cost which we denote as $C_{\mathrm{ROPE}}^{\mathrm{intra}}$. For inter-MBs with the Cauchy–Schwartz approximation, let us denote the complexity as $C_{\mathrm{ROPE}}^{\mathrm{inter}}$, where $C_{\mathrm{ROPE}}^{\mathrm{inter}} > C_{\mathrm{ROPE}}^{\mathrm{intra}}$. While ROPE complexity is not greater than ME, it is still quite substantial.

The computational cost for CQ also includes the DCT forward transform which is denoted as $C_{\mathrm{CQ}}^{\mathrm{DCT}}$. The coefficient DQ and inverse DCT transform (IDCT) have comparable complexity $C_{\mathrm{CQ}}^{\mathrm{DCT}} \simeq C_{\mathrm{DQ}}^{\mathrm{IDCT}}$. The CQ/DCT and DQ/IDCT complexities are negligible compared to ME or ROPE estimation, $C_{\mathrm{CQ}}^{\mathrm{DCT}}, C_{\mathrm{DQ}}^{\mathrm{IDCT}} \ll C_{\mathrm{ROPE}}, C_{\mathrm{ME}}$.

Rate-distortion optimization is the most complex since it makes use of the previous segments. Assuming n reference frames ($n = 2$ for our case), ME is run n times. Thus, the cost of encoding one MB is $C_{\mathrm{MB}} = n \times C_{\mathrm{ME}} + C_{\mathrm{RD}}$, where C_{RD} is the RD cost. In order to optimize for 31 possible QP parameters and $n + 1$ modes (intra and n inter), the ROPE estimation part, together with CQ/DCT and DQ/IDCT are run for $31 \times (n + 1)$ times. MC is, however, run only for $31 \times n$ times since intra modes do not require it. We obtain

$$C_{\mathrm{RD}} = (31 \times (n+1))(C_{\mathrm{ROPE}} + C_{\mathrm{CQ}}^{\mathrm{DCT}} + C_{\mathrm{DQ}}^{\mathrm{IDCT}}) + 31 \times n \times C_{\mathrm{MC}}. \quad (19)$$

Thus, R-D optimization complexity increases linearly with n, just as ME does.

Feedback: In the case of feedback, one additional part is present, decoder tracking. The encoder takes advantage of feedback ACK/NACK signals to reconstruct past frames exactly the way they were reconstructed at the decoder side (re-decoding). The intermediate frames between the re-decoded one and the current cannot be re-decoded since no feedback exists for them, but we re-derive the ROPE estimates for them using the last re-decoded frame as a starting point. If d is the feedback delay, the complexity is $C_{\mathrm{DT}} = C_{\mathrm{MC}} + d \times C_{\mathrm{ROPE}}$. We observe that it is invariant with respect to n.

B. Memory Requirements

We denote the number of pixels in the image as S. We assume grayscale images at 1 byte (unsigned char) per pixel. Assume again n reference frames.

No Feedback: The encoder needs to buffer $n + 1$ images (1 current, 1 ST and $n - 1$ LTs) that require S bytes each. In addition, we need to buffer $n + 1$ ROPE estimates which, however, are stored as floats, thus requiring $4 \times S$ bytes each (a float is stored using 4 bytes).

Feedback: Tracking past frames requires some additional buffering. The obvious buffered frames are the last acknowledged one and *its* previous one, in unsigned char format. This happens for the SF case. In the dual- and multiple-frame case, the buffering requirements multiply.

For example, in approach A in Fig. 7, consider the encoding of frame 7. Frame 2 has just been re-decoded, and we wish to use this re-decoded frame to improve the estimates of frames 3, 4, 5, and 6. First, re-decoded frames 0 and 1 must have been buffered in order to re-decode frame 2, as they were the LT and ST frames for frame 2. After re-decoding frame 2, the encoder can purge re-decoded frame 1, since that will no longer be needed. However, re-decoded frame 0 (since it is the LT frame for frame 3) must be kept until the ACK/NACK information arrives for frame 3. Re-decoded frames 0 and 2 are used to improve the estimate of frame 3. Re-decoded frame 2 and estimated frame 3 are then used to improve the estimate of frame 4. Re-decoded frame 2 and estimated frame 4 are used to improve the estimate of frame 5. Last, estimated frames 4 and 5 are used to improve the estimate of frame 6. Now the encoder can encode frame 7. So, in this example, the largest number of frames being buffered at any given time is 7 (that is, frames 0, 1, 2, 3, 4, 5, and 6) in addition to the frame to be encoded (frame 7).

In general, buffering requirements for the feedback case increase linearly with n and can be a significant impediment to implementation.

Let us now consider an example of computational and memory requirements for Fig. 7. We assume $N = 2$ and $d = 5$. As the analysis in the memory requirements subsection showed, we need to buffer eight frames (the current one, three re-decoded ones, and four estimates). If we had instead used traditional SF encoding with ROPE estimation then only four frames would need to be buffered (the current one, one estimate, the current and the previous re-decoded). Thus, we obtain a 100% increase in memory complexity for these *particular* parameters. Computational complexity also increases. C_{ME} increases by 100% and C_{RD} by 50% compared to SF. Given that C_{ME} represents roughly 65% of total complexity and C_{RD} the remaining 35%, we calculate that the increase in computational complexity when going to a dual-frame scheme is 82.5%.

C. Conclusions on Complexity

Both the memory requirements and computational requirements of using ROPE within a multiple-frame framework are large, growing linearly with n (the number of reference frames). However, past research [9], as well as our own simulations, confirmed that the performance gains grow sub-linearly with n; there is quickly a point of diminishing returns after which increasing n produces trivial or no gains. Our simulations showed an advantage of up to 0.35 dB for $n = 6$ (five LT frames) with ROPE compared to using dual frame with ROPE, for certain sequences. However, our experiments showed that most of the gain over SF ROPE is obtained through dual-frame ROPE. Since most of the performance gain can be captured by using only two reference frames, whereas the complexity grows linearly with n, we chose to use $n = 2$.

IX. Conclusion

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. At the same time, using only a single extra frame buffer keeps the computational complexity relatively low. An inter/intra mode switching algorithm coupled with the additional frame buffer provides a very robust and efficient bitstream. The experimental results showed that when feedback is employed, dual-frame schemes consistently outperform SF ones, and the advantage tends to become more apparent as the bit rate or the packet loss rate grows large. In the case where feedback is not available, dual frame when used together with the Cauchy–Schwartz approximation outperforms all other variants, and for most of the cases, the advantage is more pronounced as the packet error rate grows large.

With visual inspection of the reconstructed sequences, it is apparent that the dual-frame predictor provides a noticeably

smoother viewing experience. Background details are preserved, and packet losses generally affect only MBs with high motion, unlike in the SF reconstruction where visual distortion encompasses the entire picture.

A predetermined and fixed value of the jump updating parameter N is not optimal for all sequences. Future work will concentrate on finding good rules for choosing N and D for a general (N, D) updating scheme, and for choosing LT frames in an irregular updating scheme. It would be desirable to know which update parameter is best for a given sequence. Some sequences exhibit long-term statistics that could be best captured by using relatively large update parameters or by setting a constant distance frame buffer in the remote past. Another interesting aspect is to determine the potential in using multiple future reference frames in bi-directional prediction.

ACKNOWLEDGMENT

The authors would like to thank Prof. K. Rose and Dr. R. Zhang for making their source code available for the ROPE algorithm. They would also like to thank the anonymous reviewers for their many helpful remarks.

REFERENCES

[1] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the internet," *IEEE/ACM Trans. Networking*, vol. 7, pp. 458–472, Aug. 1999.

[2] B. Girod, M. Kalman, Y. J. Liang, and R. Zhang, "Advances in channel-adaptive video streaming," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, 2002, pp. 9–12.

[3] H. Schulzrinne, A. Rao, and R. Lanphier, "Real time streaming protocol (RTSP)," IETF, RFC 2326, 1998.

[4] T. Sikora, "The MPEG-4 video standard verification model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 19–31, Feb. 1997.

[5] G. Côté, B. Erol, M. Gallant, and F. Kossentini, "H.263+: Video coding at low bit rates," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 849–865, Nov. 1998.

[6] T. Wiegand, "Joint final committee draft for joint video specification H.264," Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-D157, 2002.

[7] M. Gothe and J. Vaisey, "Improving motion compensation using multiple temporal frames," in *Proc. IEEE Pacific Rim Conf. Communications, Computers and Signal Processing*, vol. 1, May 1993, pp. 157–160.

[8] N. Vasconcelos and A. Lippman, "Library-based image coding," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. V, Apr. 1994, pp. 489–492.

[9] T. Wiegand, X. Zhang, and B. Girod, "Long-term memory motion-compensated prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 70–84, Feb. 1999.

[10] 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, pp. 212–220, Feb. 1997.

[11] C.-S. Kim, R.-C. Kim, and S.-U. Lee, "Robust transmission of video sequence using double-vector motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 1011–1021, Sept. 2001.

[12] M. Budagavi and J. D. Gibson, "Multiframe video coding for improved performance over wireless channels," *IEEE Trans. Image Processing*, vol. 10, pp. 252–265, Feb. 2001.

[13] T. Wiegand, M. Lightstone, D. Mukherjee, T. Campbell, and S. Mitra, "Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H.263 standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 182–190, Apr. 1996.

[14] A. Schuster and A. Katsaggelos, "Fast and efficient mode and quantizer selection in the rate-distortion sense for H.263," in *Proc. SPIE Visual Communications Image Processing*, vol. 2727, 1996, pp. 784–795.

[15] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Processing Mag.*, vol. 15, pp. 23–50, Nov. 1998.

[16] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Mag.*, vol. 15, pp. 74–90, Nov. 1998.

[17] G. Côté and F. Kossentini, "Optimal intra coding of blocks for robust video communication over the internet," *Signal Process.: Image Commun., Special Issue Real-Time Video over Internet*, vol. 15, pp. 25–34, Sept. 1999.

[18] 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, pp. 966–976, June 2000.

[19] 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, pp. 977–995, June 2000.

[20] T. Wiegand, N. Färber, K. Stuhlmüller, and B. Girod, "Long-term memory motion-compensated prediction for robust video transmission," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, 2000, pp. 152–155.

[21] ——, "Error-resilient video transmission using long-term memory motion compensated prediction," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 1050–1062, June 2000.

[22] T. Stockhammer, T. Wiegand, and S. Wenger, "Optimized transmission of H.26L/JVT coded video over packet-lossy networks," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, 2002, pp. 173–176.

[23] 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, pp. 952–965, June 2000.

[24] S. Cen and P. Cosman, "Comparison of error concealment strategies for MPEG video," in *Proc. IEEE Wireless Communications and Networking Conf.*, vol. 1, 1999, pp. 329–333.

[25] P. Salama, N. B. Shroff, and E. J. Delp, "Error concealment in MPEG video streams over ATM networks," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 1129–1144, June 2000.

[26] J. Choi and D. Park, "A stable feedback control of the buffer state using the controlled Lagrange multiplier method," *IEEE Trans. Image Processing*, vol. 3, pp. 546–558, Sept. 1994.

[27] "Video coding for low bit rate communication," ITU-T Recommendation H.263 Version 2 ("H.263+"), 1998.

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

Athanasios Leontaris (S'97) received the Diploma degree in electrical and computer engineering from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2000 and the M.S. degree in electrical engineering from the University of California at San Diego (UCSD), La Jolla, in 2002. He is currently pursuing the Ph.D. degree at the Information Coding Laboratory, Department of Electrical and Computer Engineering, UCSD.

His research interests include image processing, compression of images and three-dimensional medical volumes, video compression and transmission over lossy packet networks, and wireless channels.

Pamela C. Cosman (S'88–M'93–SM'00) received the B.S. degree (with honors) in electrical engineering from the California Institute of Technology, Pasadena, in 1987, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1989 and 1993, respectively.

She was an NSF Postdoctoral Fellow at Stanford University and a Visiting Professor at the University of Minnesota, Minneapolis, from 1993 to 1995. Since July 1995, she has been with the faculty of the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, where she is currently an Associate Professor. Her research interests are in the areas of image and video compression and processing.

Dr. Cosman is a member of Tau Beta Pi and Sigma Xi. She is the recipient of the ECE Departmental Graduate Teaching Award (1996), a Career Award from the National Science Foundation (1996 to 1999), and a Powell Faculty Fellowship (1997 to 1998). She was an Associate Editor of the IEEE COMMUNICATIONS LETTERS from 1998 to 2001, a Guest Editor of the June 2000 special issue of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS on "error-resilient image and video coding," and the Technical Program Chair of the 1998 Information Theory Workshop, San Diego. She is currently an Associate Editor of the IEEE SIGNAL PROCESSING LETTERS and a Senior Editor of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS.