Video Compression with Intra/Inter Mode Switching and a Dual Frame Buffer *

Athanasios Leontaris and Pamela C. Cosman
University of California, San Diego

Abstract

Video codecs that use motion compensation have recently achieved performance improvements from the use of intra/inter mode switching decisions 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. Also, we improve the mode switching algorithm with the use of half-pel motion vectors.

1 Introduction

Most of today's 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 [1], H.263 [10] and the latest and state-of-the-art H.26L. The idea of using more than one past reference frame to improve coding efficiency is not new. The first mention [3] of multiple reference frames dates almost a decade back; it was shown that the mean-squared error (MSE) between the current frame and the predicted one strictly decreases by using multiple temporal frames for motion compensation. Another early attempt to code an image using a so-called *library* of past frame components can be found in [2], and made use of vector quantization. Multiple frame prediction was also treated in [8].

In [9], 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. One frame was the previous one, as in many hybrid codecs, and the second one contained a reference

*Supported in part by the National Science Foundation, the Center for Wireless Communications at UCSD, and the CoRe program of the State of California. The authors are with the Department of Electrical and Computer Engineering, University of California, San Diego, 9500 Gilman Dr. MC 0407, La Jolla, CA 92093-0407. Email: {pcosman, aleontar}@code.ucsd.edu

Proceedings of the Data Compression Conference (DCC'03)
1068-0314/03 $17.00 © 2003 IEEE

frame from the more distant past that was periodically updated according to a predefined rule. In [7], the authors use a linear weighted combination of two frames, primarily to enhance the error robustness of the codec. Error robustness was also studied within a multiple-reference frame scheme in [4] and [6].

As these papers showed, there is a significant gain in reconstructed PSNR to be obtained, at the expense of increased computational burden and memory complexity. Motion estimation is the main performance bottleneck in a hybrid video coding system, and can account for more than 80-90% of the total encoding time. Thus adding even one additional frame buffer can double the encoding time. The same is true with memory requirements, where the increase is also linear and thus prohibitive as the number of reference frames grows large. Some efforts were directed to finding fast and efficient algorithms for motion estimation, but some performance penalty is sustained as it is traded-off for reduced computational complexity [14].

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. The paper is organized as follows. In Section 2 we review the ROPE algorithm [11] for intra/inter mode switching. In Section 3, we show how this algorithm can be used in the context of a dual frame buffer. The use of half-pel motion vectors is covered in Section 4. Results and conclusions are presented in Section 5.

2 Baseline ROPE Algorithm

Recent attempts to switch coding modes according to error robustness criteria can be found in [12] and [11]. Our work makes use of the recursive optimal per-pixel estimate (ROPE) algorithm [11] which provides mode decisions 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. The problem that arises is how to optimally select between intra- and inter-coding for a particular macroblock, such that both error resilience and coding efficiency are achieved.

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 (MBs) and is transmitted as a single packet. Each packet can be independently received and decoded, due to resynchronization markers. Thus, a 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. The decoder replaces the lost macroblock by one from the previous frame, using as motion vector (MV) the median of the MVs of the three closest macroblocks 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 macroblock is replaced with the co-located one from the previous frame.

We will now summarize the ROPE algorithm [11] in some detail as these equations will

prove useful in elaborating our proposed method. 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:

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

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 [11], 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 motion vector 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 [11]:

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

$$E\{(\tilde{f}_n^i)^2\} = (1-p)(\hat{f}_n^i)^2 + p(1-p)E\{(\tilde{f}_{n-1}^k)^2\} + p^2 E\{(\tilde{f}_{n-1}^i)^2\} \tag{3}$$

For an inter-coded MB, let us assume that its true motion vector 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:

$$\hat{f}_n^i = \hat{f}_{n-1}^j + \hat{e}_n^i \tag{4}$$

The encoder transmits $\hat{e}_n^i$ and the MB's motion vector. 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:

$$\tilde{f}_n^i = \tilde{f}_{n-1}^j + \hat{e}_n^i \tag{5}$$

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 moment of $\tilde{f}_n^i$ for a pixel in an inter-coded MB is then given by:

$$E\{\tilde{f}_n^i\} = (1-p)\left(\hat{e}_n^i + E\{\tilde{f}_{n-1}^j\}\right) + p(1-p)E\{\tilde{f}_{n-1}^k\} + p^2 E\{\tilde{f}_{n-1}^i\} \tag{6}$$

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

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.

2.1 Rate-Distortion Framework

The ROPE algorithm takes into account the expected distortion due to both compression and transmission errors for optimal mode switching. The encoder switches between intra- or inter-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 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 macroblock basis. Therefore, the encoding mode for each MB is chosen by minimizing

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

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 [13]. Both the *coding mode* and the *quantization step size* 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 an unmodified hybrid video encoder.

We note that while the ROPE algorithm is optimal under the given assumptions, there is potential for improvement by incorporating the motion vector choice into the rate-distortion framework, by correctly estimating distortion for half-pel vectors (the algorithm only models distortion for integer motion vectors) or by taking into account the fact that all pixels in a GOB are either lost together, or transmitted correctly.

3 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 varies from as recent as frame $n-2$ to as old as frame $n-N$. When the long term frame is $n-N$, then, when the encoder moves on to encoding frame $n+1$, the short term reference frame will move forward one to frame n, and the long term reference frame will jump forward to be frame $n-1$. The long term reference frame will then remain static for N frames, and then jump forward again.

Each macroblock 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). 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 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.

We now describe 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 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 Equation 1.

To compute the moments in Equation 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 motion vector 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 motion vector 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,

$$E\{\tilde{f}^i_n\} = (1-p)\left(\hat{e}^i_n + E\{\tilde{f}^j_{n-l}\}\right) + p(1-p)E\{\tilde{f}^k_{n-1}\} + p^2 E\{\tilde{f}^i_{n-1}\} \tag{11}$$

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

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 motion vectors may be highly uncorrelated. If the upper GOB is also lost, then we conceal the MB using the co-located block from the previous frame.

The presence of neighboring uncorrelated motion vectors negatively affects motion vector coding efficiency. There is a bit rate loss due to inaccurate prediction of motion vectors from their neighboring motion vectors. Furthermore, compression efficiency is reduced because we use 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.

Since the quantization parameter QP takes values from 1 to 31, the original ROPE algorithm optimizes over 62 potential combinations of coding modes (intra or inter) and quantization parameters. 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.

4 Half-Pixel Accuracy Extension to the ROPE Algorithm

The use of integer motion vectors limits the reference choices in the previous frame. Most video codecs show a performance advantage when half-pel motion vectors 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 motion vector 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 motion vectors, but to leave it out of the ROPE distortion calculation altogether. This is what is done in [11], and it provides some improvement over the use of strictly integer motion vectors. 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 (EC) is still done using only the integer portion of the motion vectors, and therefore Equations 2 and 3 for the intra-coded MBs are unchanged. Returning to Equations 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.

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. The first moment is computationally tractable:

$$E\{\tilde{f}_{n-1}^j\} = \frac{1}{2}\left[1 + E\{\tilde{f}_{n-1}^{k_1}\} + E\{\tilde{f}_{n-1}^{k_2}\}\right] \tag{13}$$

But the expression for the second moment is:

$$E\{(\tilde{f}^j_{n-1})^2\} = \frac{1}{4}\Big[1 + E\{(\tilde{f}^{k_1}_{n-1})^2\} + E\{(\tilde{f}^{k_2}_{n-1})^2\} + 2E\{\tilde{f}^{k_1}_{n-1}\} + 2E\{\tilde{f}^{k_2}_{n-1}\} + 2E\{\tilde{f}^{k_1}_{n-1}\tilde{f}^{k_2}_{n-1}\}\Big] \quad (14)$$

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. We will approximate it using the cosine inequality:

$$E\{(\tilde{f}^j_{n-1})^2\} \leq \frac{1}{4}\Big[1 + E\{(\tilde{f}^{k_1}_{n-1})^2\} + E\{(\tilde{f}^{k_2}_{n-1})^2\} + 2E\{\tilde{f}^{k_1}_{n-1}\} + 2E\{\tilde{f}^{k_2}_{n-1}\} + 2\sqrt{E\{(\tilde{f}^{k_1}_{n-1})^2\}E\{(\tilde{f}^{k_2}_{n-1})^2\}}\Big] \quad (15)$$

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:

$$E\{\tilde{f}^j_{n-1}\} = \frac{1}{4}\Big[2 + E\{\tilde{f}^{k_1}_{n-1}\} + E\{\tilde{f}^{k_2}_{n-1}\} + E\{\tilde{f}^{k_3}_{n-1}\} + E\{\tilde{f}^{k_4}_{n-1}\}\Big] \quad (16)$$

The accurate but intractable expression for the second moment is:

$$\begin{aligned} E\{(\tilde{f}^j_{n-1})^2\} = \frac{1}{16}\Big[&4 + E\{(\tilde{f}^{k_1}_{n-1})^2\} + E\{(\tilde{f}^{k_2}_{n-1})^2\} + E\{(\tilde{f}^{k_3}_{n-1})^2\} + E\{(\tilde{f}^{k_4}_{n-1})^2\} \\ &+ 4E\{\tilde{f}^{k_1}_{n-1}\} + 4E\{\tilde{f}^{k_2}_{n-1}\} + 4E\{\tilde{f}^{k_3}_{n-1}\} + 4E\{\tilde{f}^{k_4}_{n-1}\} \\ &+ 2E\{\tilde{f}^{k_1}_{n-1}\tilde{f}^{k_2}_{n-1}\} + 2E\{\tilde{f}^{k_1}_{n-1}\tilde{f}^{k_3}_{n-1}\} + 2E\{\tilde{f}^{k_1}_{n-1}\tilde{f}^{k_4}_{n-1}\} \\ &+ 2E\{\tilde{f}^{k_2}_{n-1}\tilde{f}^{k_3}_{n-1}\} + 2E\{\tilde{f}^{k_2}_{n-1}\tilde{f}^{k_4}_{n-1}\} + 2E\{\tilde{f}^{k_3}_{n-1}\tilde{f}^{k_4}_{n-1}\}\Big] \end{aligned} \quad (17)$$

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

$$\begin{aligned} E\{(\tilde{f}^j_{n-1})^2\} \leq \frac{1}{16}\Big[&4 + E\{(\tilde{f}^{k_1}_{n-1})^2\} + E\{(\tilde{f}^{k_2}_{n-1})^2\} + E\{(\tilde{f}^{k_3}_{n-1})^2\} + E\{(\tilde{f}^{k_4}_{n-1})^2\} \\ &+ 4E\{\tilde{f}^{k_1}_{n-1}\} + 4E\{\tilde{f}^{k_2}_{n-1}\} + 4E\{\tilde{f}^{k_3}_{n-1}\} + 4E\{\tilde{f}^{k_4}_{n-1}\} \\ &+ 2\sqrt{E\{(\tilde{f}^{k_1}_{n-1})^2\}E\{(\tilde{f}^{k_2}_{n-1})^2\}} + 2\sqrt{E\{(\tilde{f}^{k_1}_{n-1})^2\}E\{(\tilde{f}^{k_3}_{n-1})^2\}} \\ &+ 2\sqrt{E\{(\tilde{f}^{k_1}_{n-1})^2\}E\{(\tilde{f}^{k_4}_{n-1})^2\}} + 2\sqrt{E\{(\tilde{f}^{k_2}_{n-1})^2\}E\{(\tilde{f}^{k_3}_{n-1})^2\}} \\ &+ 2\sqrt{E\{(\tilde{f}^{k_2}_{n-1})^2\}E\{(\tilde{f}^{k_4}_{n-1})^2\}} + 2\sqrt{E\{(\tilde{f}^{k_3}_{n-1})^2\}E\{(\tilde{f}^{k_4}_{n-1})^2\}}\Big] \end{aligned} \quad (18)$$

and we use this upper limit to approximate the second moment.

5 Results

We modified an existing H.263+ video codec [10] in two ways. In the case of single-frame (SF) motion compensation, we used the ROPE algorithm for mode switching decisions. The resulting bitstream is fully compliant with the H.263+ standard [10]. Secondly, we modified the H.263+ codec to make use of one additional (long-term) frame buffer. This frame buffer was periodically updated according to an *update parameter* N as in [9]. For both the single frame and dual frame cases, we measured the performance for integer and half-pel motion vectors. 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 half-pel and accurate half-pel cases.

In our experiments, N was set to 4, 3, and 2, for 30, 15, and 10 frames per second, respectively. N was kept small to increase MV correlation, and thus improve MV coding efficiency. The GOB-packet error probability was tested with values of $p = 0$, 0.05, 0.10, 0.15 and 0.20. The resulting dual frame encoder is not standard compliant, 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. We tested various bitrates ranging from 64kbps to 400kbps. The results shown have been averaged over 25 runs using random error patterns. The same error patterns were used for all algorithm versions.

Figure 1(a) shows the performance of the proposed ROPE variants for the "Carphone" QCIF image sequence at 15fps for an error rate of $p = 10\%$. Half-pel versions outperform the equivalent integer-pel ones by a significant margin. The accurate half-pel versions outperform the regular half-pel variants. The dual frame buffer version exhibits a higher PSNR performance of roughly 0.4dB on average compared to the single frame case. Similar trends appear in Figure 1(b), which contains results from the QCIF image sequence "Container" at 10fps for a GOB error rate of $p = 10\%$. In this case, the performance delta from the baseline ROPE half-pel scheme to accurate half-pels is large, roughly 0.6dB. The dual frame buffer version is still able to produce the most efficient bit stream.

Fig. 2(a) shows PSNR performance for GOB error rates from $p = 0\%$ to $p = 20\%$. Performance degrades significantly when the probability of packet drops reaches 20%. Almost 4dB are lost in peak SNR. The dual frame buffer case proves its potential particularly in the higher error cases, providing a significant gain of roughly 0.5dB. For $p = 5\%$, it still outperforms the single frame case, but the difference shrinks to around 0.3dB. It is also interesting to note that for $p = 20\%$ the performance of the three single-frame variants converges. The same is true for the dual frame buffer variants. At $p = 20\%$, the errors are so severe that little can be gained by using half-pels, whether or not they are accurately modeled. However, the dual frame buffer case still provides an advantage over the SF one. Fig. 2(b) shows similar trends to Fig. 2(a) but the gains from the dual frame are larger. Again the single-frame variants exhibit a slightly higher rate of descent due to increasing GOB error rate than do the dual frame versions. Results obtained for 30 fps were found to be comparable to those for 10 and 15 fps.

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

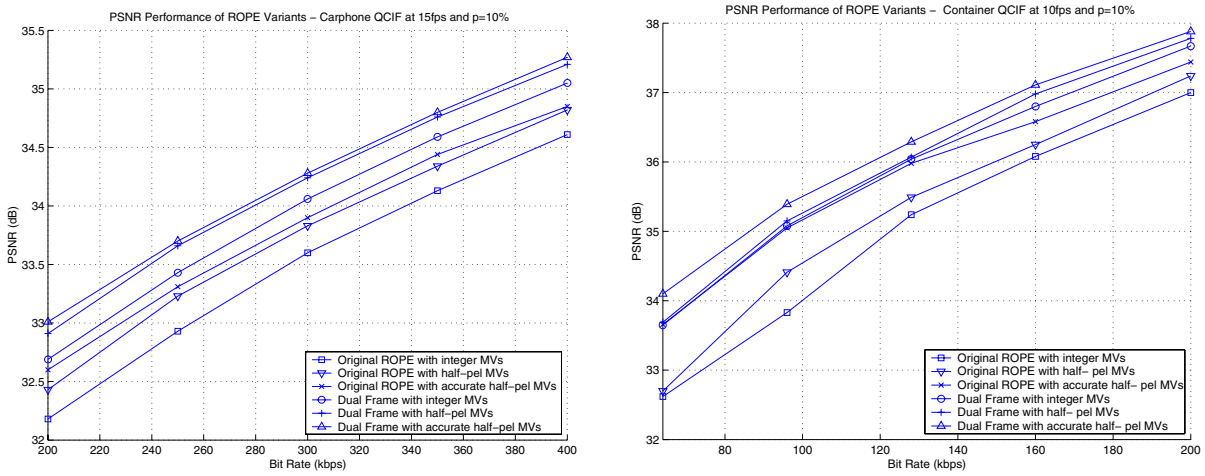


Figure 1: PSNR performance versus bit rate for GOB error rate $p = 10\%$.

less correlated, and the respective motion vectors have generally higher and more variable 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.

It is apparent that the addition of the long-term frame buffer improves the encoder's compression efficiency and renders the bitstream more robust to packet drops. However, for certain sequences the effect is small, and it depends on the update parameter N. A fixed N is not optimal for all sequences. It would be desirable to know which update parameter is best for a given sequence.

In conclusion, experimental results proved that the addition of a single long-term reference frame buffer can improve the compression performance of a hybrid video codec when used with an optimal mode switching scheme. The performance gain varies according to the statistics of the underlying image sequence.

References

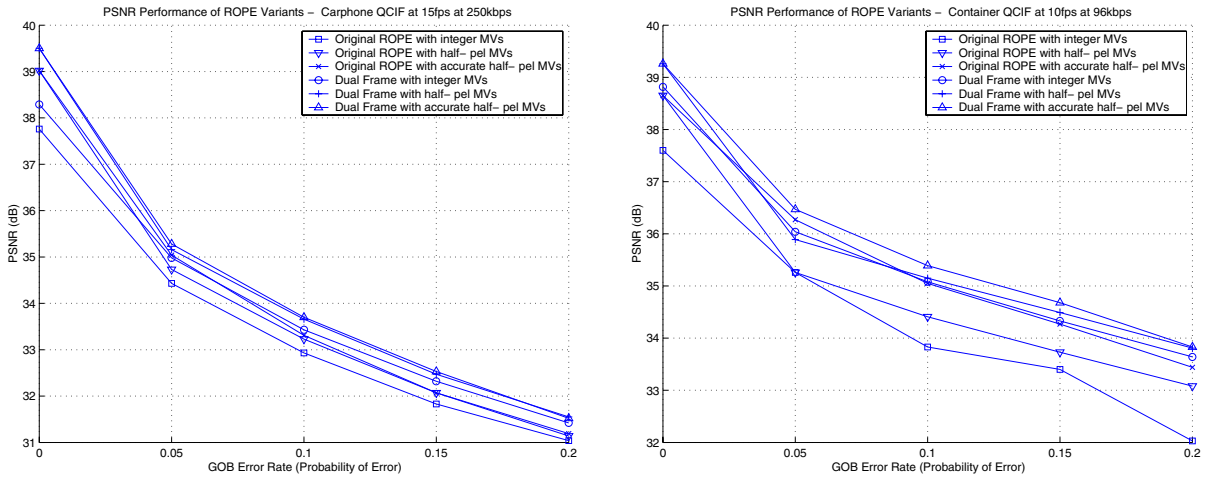
[1] T. Sikora, "The MPEG-4 Video Standard Verification Model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 19-31, February 1997.

[2] N. Vasconcelos and A. Lippman, "Library-based Image Coding," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. v, pp. V/489-V/492, 1994.

[3] M. Gothe and J. Vaisey, "Improving Motion Compensation Using Multiple Temporal Frames," *IEEE Pac. Rim Conf. on Comm., Comp. and Sig. Proc.,* v.1, pp. 157-160, 1993.

[4] T. Wiegand, N. Färber, K. Stuhlmüller, and B. Girod, "Long-Term Memory Motion-Compensated Prediction for Robust Video Transmission," *ICIP*, vol. 2, pp. 152-155, 2000.

[5] S. Cen and P. Cosman, "Comparison of Error Concealment Strategies for MPEG Video," *IEEE Wireless Comm. and Networking Conference*, vol. 1, pp. 329-333, 1999.



(a) (b)

Figure 2: PSNR performance versus GOB error rate.

[6] M. Budagavi and J. D. Gibson, "Multiframe Video Coding for Improved Performance Over Wireless Channels," *IEEE Trans. Image Proc.*, vol. 10, no. 2, pp. 252-265, February 2001.

[7] C.-S. Kim, R.-C. Kim, and S.-U. Lee, "Robust Transmission of Video Sequence Using Double-Vector Motion Compensation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 9, pp. 1011-1021, September 2001.

[8] T. Wiegand, X. Zhang, and B. Girod, "Long-Term Memory Motion-Compensated Prediction," *IEEE Trans. Circ. and Systems for Video Techn.*, vol. 9, no. 1, pp. 70-84, Feb. 1999.

[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. Circ. and Systems for Video Techn.*, vol. 7, no. 1, pp. 212-220, Feb. 1997.

[10] G. Côté, B. Erol, M. Gallant and F. Kossentini, "H.263+: Video Coding at Low Bit Rates," *IEEE Trans. Circ. and Systems for Video Techn.*, vol. 8, no. 7, pp. 849-865, Nov. 1998.

[11] R. Zhang, S. L. Regunathan, and K. Rose, "Video Coding with Optimal Inter/Intra-Mode Switching for Packet Loss Resilience," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 966-976, June 2000.

[12] G. Côté, S. Shirani, and F. Kossentini, "Optimal Mode Selection and Synchronization for Robust Video Communications over Error-Prone Networks," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 952-965, June 2000.

[13] J. Choi and D. Park, "A Stable Feedback Control of the Buffer State Using the Controlled Lagrange Multiplier Method," *IEEE Trans. Image Proc.*, vol. 3, pp. 546-558, Sep. 1994.

[14] H.-S. Wang and R. M. Mersereau, "Fast Algorithms for the Estimation of Motion Vectors," *IEEE Trans. Image Processing*, vol. 8, no. 3, pp. 435-438, March 1999.