Drift-Resistant SNR Scalable Video Coding

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

***Abstract*—We address the problem of enhancement layer drift estimation for fine granular scalable video. An optimal per-pixel drift estimation algorithm is introduced. The encoder assumes that there is some truncation of the enhancement layer, which does not allow the enhancement layer reference to be properly reconstructed, and the encoder recursively estimates the associated drift and chooses coding modes accordingly. The approach yields performance gains of about 1 dB across low to medium rates. In addition, we investigate dual frame prediction, for both base and enhancement layer, with pulsed-quality allocation in the base layer.**

***Index Terms*—Bitplane coding, fine granularity scalability, H.264, H.26L, multiple frame prediction, pulsed quality, scalable video coding, video compression.**

I. Introduction

FINE granular scalable (FGS) video coding has emerged as an important research topic in recent years. Instead of compressing for a given target rate, it is desirable to compress for a range of bit rates at which the sequence can be potentially decoded. This is critical for internet video streaming, because there is usually no guarantee of constant bandwidth. One can extract multiple versions of the same video, at different levels of quality, from a single compressed file, and then stream them to recipients with different bit rate requirements. FGS was recently accepted for inclusion into the state-of-the-art scalable video codec jointly developed by ISO and ITU-T [2]. The first standardized effort on FGS video coding was the MPEG-4 FGS signal-to-noise ratio (SNR) scalability extension [3]. The base layer consists of a standard single-layer MPEG-4 bitstream while the enhancement layer (EL) is coded with the bitplane technique and references only the base layer reconstruction of the image. Bitplane coding provides a completely embedded stream that can be arbitrarily truncated to fit the available bandwidth.

In [4], Wu *et al.* introduced progressive fine granularity scalability (PFGS), which uses an additional EL reference frame to improve motion prediction. Assuming availability of the base layer and EL references, the frames being encoded alternate between those two layers as reference. In [5], performance was improved by selecting the reference layer on a macroblock basis, called MB-PFGS. He *et al.* [6] combined H.264/AVC with MB-PFGS to produce a scalable coder that outperformed MPEG-4 FGS, using both base and EL information during motion estimation. PFGS suffers from drift due to possible loss of the previous EL. A drift estimation technique was proposed in [7]. The drift was not modeled probabilistically, hence, could not be used to estimate first or higher order moments of the enhancement reference pixels.

To further reduce drift and improve compression, we investigate incorporating multiple frame prediction into FGS scalable video coding. The earliest attempt is found in [8] which used the previous five frames as additional references. Another approach to multiple references is found in [9]. Two frames (one is the short term) are buffered and reference frame selection is biased in favor of the farthest frame. While all frames serve as references for their immediate subsequent frame, a subset of frames are retained in the frame memory for reference by later frames. A separate approach with multiple references that makes use of leaky prediction to constrain drift was presented in [10], where the drift error was modeled as the worst possible.

In this paper, we apply pulsed-quality allocation to periodically updated long-term frames used for dual frame prediction as proposed in [11]. Uneven quality allocation is applied only to the base layer. In dual frame prediction, two reference frames are employed, short and long term. The reference frame is selected to minimize distortion. The paper is organized as follows. Section II gives an overview of the EL coding modes, and describes our algorithm for optimal per-pixel estimation. Section III discusses the implementation of the recursive estimation and Section IV presents the dual frame prediction scheme. Experimental results are presented and discussed in Section V. The paper concludes in Section VI.

II. Optimal Per-Pixel Estimation of Drift

Base layer macroblocks (MBs) are encoded with one of the many possible modes defined in the H.264/AVC standard. For the EL, however, every MB can be encoded with three possible coding modes [Fig. 1(a)] [5]. The top dark gray squares denote base layers, bottom light gray squares denote enhancement references, and white squares with dashed lines denote partially decoded (top) or higher (bottom) enhancement layers. Base layer MBs are always reconstructed exclusively from previous base layers. Black arrows denote prediction, while white arrows denote reconstruction. We note that hereon "prediction" refers to the motion compensated (MC) prediction at the encoder side, while "reconstruction" stands for the MC prediction at the decoder side.

The first coding mode is *LPLR*, where an enhancement MB is predicted and reconstructed from the previous base layer. Using this mode, and assuming that the base layer is always received in its entirety, no prediction/reconstruction mismatch is possible

Manuscript received May 27, 2005; revised October 19, 2005. This work was supported in part by the National Science Foundation; in part by the Center for Wireless Communications at the University of California, San Diego; in part by the Office of Naval Research; and in part by the UC Discovery Grant program of the State of California. An early version of the per-pixel estimate algorithm presented here appeared in [1]. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Fernando M. B. Pereira.

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

Digital Object Identifier 10.1109/TIP.2006.877412

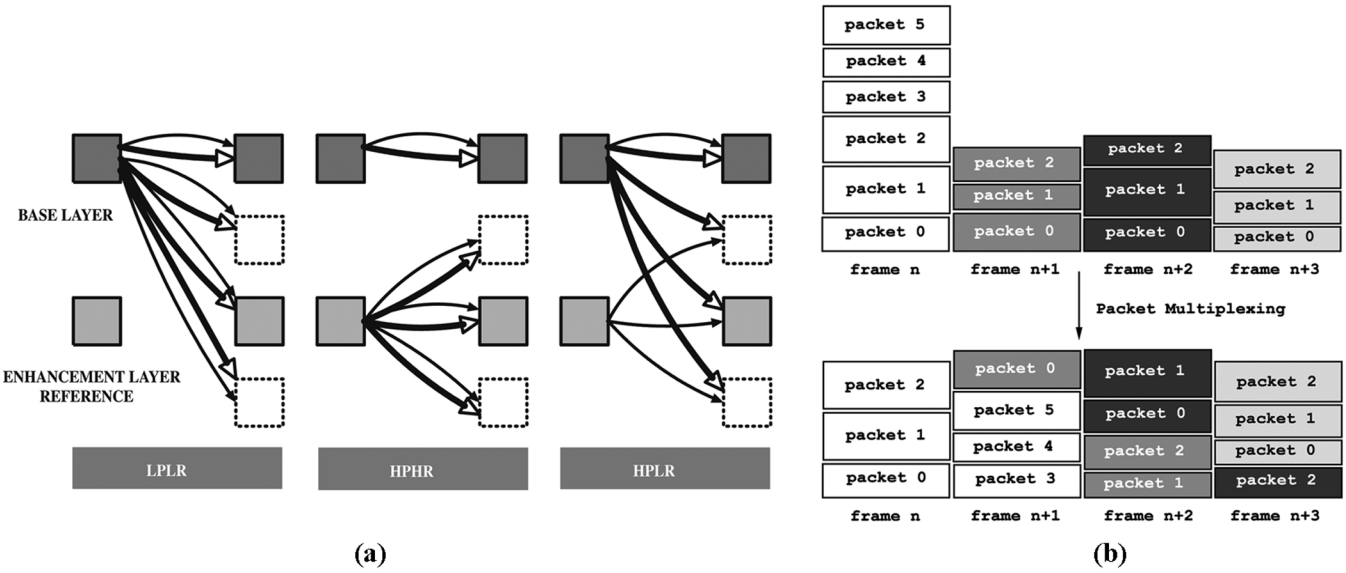

Fig. 1. (a) Enhancement layer coding modes. (b) Bitstream generation and transmission with delay in pulsed-quality framework.

and drift from previous frames is stopped. The coding efficiency is degraded due to the low quality motion compensation and reference.

The two other coding modes involve prediction from the EL reference. In *HPHR*, the enhancement MB is both predicted and reconstructed from the EL reference. This yields high compression, provided the previous enhancement reference was received in its entirety. If not, we have drift. To counter this, in *HPLR* mode, prediction still takes place from the enhancement reference, but reconstruction now uses the previous base layer. The quality is lower than HPHR, but drift is contained. At the decoder side, the modes "LPLR" and "HPLR" are identical, since, in both modes, the base layer reference is used for reconstruction. Thus, only one bit is needed to signal an enhancement layer mode.

Hence, selecting HPHR provides best quality with drift, LPLR yields low quality without drift, while HPLR is a tradeoff between those two. Leaky prediction [12] uses as a prediction reference a weighted superposition of the EL and BL predictions. Quality is a tradeoff, and, while drift exists, it attenuates to zero over time provided the EL weighting is sufficiently small. In our scheme, the suppression of drift is a problem of coding mode decision.

Let n be the number of the current frame, and (i,j) the spatial coordinates of the pixel we seek to estimate. The motion vector (MV) that points to the prediction block in frame $n-1$ is denoted (v_x, v_y). Let $(i+v_x, j+v_y) = (\alpha, \beta)$. Let f_k denote the probability that the received EL portion has been truncated at rate R_k (i.e., available bandwidth at a particular moment is R_k), for $k=0$ to $N-1$, where $R_l < R_k$ for $l<k$, and N is the number of operational rates. Let R_{ER} denote the enhancement reference rate. Even if rate $R > R_{ER}$ is available to the decoder, the enhancement reference will still be decoded at rate R_{ER}. The frame decoded at rate R will be used only for display purposes by the decoder. It is left out of the decoding loop. Disregarding the effects of the loop filter and quarter-pixel accurate motion compensation used in baseline H.264/AVC, we observe that, at the decoder, a reconstructed EL reference pixel $\tilde{p}^n_{er}(i,j)$ at frame n and spatial coordinates (i,j) can be written for LPLR and HPLR modes as

$$\tilde{p}^n_{er}(i,j) = p^{n-1}_b(\alpha,\beta) + \tilde{r}^n(i,j) \tag{1}$$

where $p^{n-1}_b(\alpha,\beta)$ is a motion-compensated base layer pixel of frame $n-1$, which is a deterministic value known by both encoder and decoder, since the BL is assumed to be received in full. Term $\tilde{r}^n(i,j)$, the reconstructed residue from the received part of the EL, can vary according to channel conditions and, thus, has to be modeled, by the encoder, as a random variable. This residue differs for LPLR and HPLR because of separate references, though the equations are unaffected. For HPHR, we obtain

$$\tilde{p}^n_{er}(i,j) = \tilde{p}^{n-1}_{er}(\alpha,\beta) + \tilde{r}^n(i,j). \tag{2}$$

Term $\tilde{p}^{n-1}_{er}(\alpha,\beta)$ is the motion-compensated pixel in the EL reference of frame $n-1$, which has to be considered random by the encoder, since the encoder cannot know if the received portion of the EL was enough to reconstruct the enhancement reference frame in full. We seek the expected values (*first moments*) of these random variables. Due to space constraints, we derive this only for HPHR

$$\begin{aligned} E\{\tilde{p}^n_{er}(i,j)\} &= E\{\tilde{p}^{n-1}_{er}(\alpha,\beta) + \tilde{r}^n(i,j)\} \\ &= E\{\tilde{p}^{n-1}_{er}(\alpha,\beta)\} + E\{\tilde{r}^n(i,j)\}. \end{aligned} \tag{3}$$

If the last term, the residual, is calculated, then our recursive estimate is complete. We use l to denote that value among the possible truncation rates where $R_{l-1} < R_{ER} \le R_l$, and obtain

$$E\{\tilde{r}^n(i,j)\} = \sum_{k=0}^{l-1} f_k r^n_k(i,j) + r^n_{ER}(i,j) \sum_{k=l}^{N-1} f_k \tag{4}$$

where $r_k^n(i,j)$ denotes the enhancement residue truncated at rate R_k, and $r_{ER}^n(i,j)$ the enhancement residue required to reconstruct the enhancement reference in full. For $k \geq l$, we set $r_k^n(i,j) = r_{ER}^n(i,j)$ since the truncated rate is enough to fully recover the enhancement reference. Per-pixel recursive estimation was previously shown to be effective in packet loss scenarios [13]. However, one needs the *second moment* of the random variable as well, to calculate the mean *squared* error during mode decision. From (2)

$$\begin{aligned} E\left\{\left(\tilde{p}_{er}^n(i,j)\right)^2\right\} &= E\left\{\left(\tilde{p}_{er}^{n-1}(\alpha,\beta) + \tilde{r}^n(i,j)\right)^2\right\} \\ &= E\left\{\left(\tilde{p}_{er}^{n-1}(\alpha,\beta)\right)^2\right\} + E\{(\tilde{r}^n(i,j))^2\} \\ &\quad + 2E\left\{\tilde{p}_{er}^{n-1}(\alpha,\beta)\tilde{r}^n(i,j)\right\}. \end{aligned} \tag{5}$$

To obtain the third term, we *assume* that prediction reference $\tilde{p}_{er}^{n-1}$ is *uncorrelated* with the residue $\tilde{r}^n$

$$\begin{aligned} E\left\{\left(\tilde{p}_{er}^n(i,j)\right)^2\right\} &= E\left\{\left(\tilde{p}_{er}^{n-1}(\alpha,\beta)\right)^2\right\} + E\{(\tilde{r}^n(i,j))^2\} \\ &\quad + 2E\left\{\tilde{p}_{er}^{n-1}(\alpha,\beta)\right\} E\{\tilde{r}^n(i,j)\}. \end{aligned} \tag{6}$$

The second moment of the residual is

$$E\left\{\left(\tilde{r}^n(i,j)\right)^2\right\} = \sum_{k=0}^{l-1} f_k\left(r_k^n(i,j)\right)^2 + \left(r_{ER}^n(i,j)\right)^2 \sum_{k=l}^{N-1} f_k. \tag{7}$$

Using (3) and (4), we recursively estimate the first moment, and with (6) and (7), we estimate the second moment for HPHR blocks. For LPLR and HPLR, the residual estimates (4) and (7) remain the same. For the first moment instead of (3), we write

$$\begin{aligned} E\left\{\tilde{p}_{er}^n(i,j)\right\} &= E\left\{p_b^{n-1}(\alpha,\beta) + \tilde{r}^n(i,j)\right\} \\ &= p_b^{n-1}(\alpha,\beta) + E\{\tilde{r}^n(i,j)\} \end{aligned} \tag{8}$$

and for the second moment instead of (6), we use

$$\begin{aligned} E\left\{\left(\tilde{p}_{er}^n(i,j)\right)^2\right\} &= \left(p_b^{n-1}(\alpha,\beta)\right)^2 + E\{(\tilde{r}^n(i,j))^2\} \\ &\quad + 2p_b^{n-1}(\alpha,\beta)E\{\tilde{r}^n(i,j)\}. \end{aligned} \tag{9}$$

These equations are used at the encoder to estimate drift optimally. This algorithm is called drift estimate per-pixel (DEPP).

III. Drift Estimate Algorithm Implementation

Mode selection for the EL is accomplished as in [5]. Instead of employing the intact enhancement reference, we use our recursive per-pixel estimates. Let $h^n(i,j)$ denote a pixel in the original current frame n at position (i,j). Let $r_e^n(i,j) = h^n(i,j) - p_{er}^{n-1}(\alpha,\beta)$ denote the prediction residual from the EL reference, and $r_b^n(i,j) = h^n(i,j) - p_b^{n-1}(\alpha,\beta)$ denote the prediction residual from the base layer. Term p_{er}, without the tilde, is the *intact* EL reference, and not an estimate. We now disregard frame indices and spatial coordinates to simplify notation. The base layer codec quantizes r_b and sends the quantized $\hat{r}_b$ to the receiver. In [5], the coding mode is selected as LPLR over either HPLR or HPHR, if

$$\|r_b - \hat{r}_b\| < \|\tilde{r}_e - \hat{r}_b\|. \tag{10}$$

The DCT residues encoded in the enhancement layer are $r_b - \hat{r}_b$ for the LPLR mode, and $r_e - \hat{r}_b$ for either HPHR or HPLR. We calculate $\tilde{r}_e = h - E\{\tilde{p}_{er}\}$ using our per-pixel estimates. Since our estimate $E\{\tilde{p}_{er}\}$ is going to be worse than the actual EL reference prediction p_{er}, doing this will slightly bias in favor of the LPLR mode.

If either HPLR or HPHR mode was selected for the EL block, we follow the approach in [5] and select HPHR over HPLR when the following inequality is satisfied

$$\|h - p_{er}\| \times c < \|p_b - p_{er}\| \tag{11}$$

where c is a constant that is fine-tuned empirically. Equation (11) trades off distortion (left side) for possible drift (right side). In this expression from [5], we replace p_{er} with the estimated predictions p_b^{n-1} or $E\left\{\tilde{p}_{er}^{n-1}\right\}$, depending on the EL coding mode. The encoder takes $N = 1$, so $f_0 = 1$, meaning that only one truncation rate R_0 is assumed to occur, and that rate is assumed to be insufficient for proper reconstruction of the enhancement layer reference $R_k < R_{ER}$. We finally note that $\|.\|$ denotes mean-squared error (MSE); hence, the need to obtain the second moments of our estimates.

We recursively estimate the EL references with (3), (4), and (8) (first moment) and (6), (7), and (9) (second moment). During mode selection, we only use the estimated predictions p_b^{n-1} and $E\left\{\tilde{p}_{er}^{n-1}\right\}$ and do not add the partial residue. Only after the EL bitstream has been fully produced, we update the estimates using the above mentioned equations, in contrast with the ROPE packet loss estimation algorithm [13] that uses the current estimates for mode selection. Due to the scalable nature of our codec this is not feasible, since the calculation of the current estimates requires the truncation of the enhancement layer under construction, and every single enhancement mode decision we make changes the way the final layer will look. We instead employ the predictions from the previous estimated reference. More complex implementations of our approach are possible if we know additional statistics (additional and more accurate f_k values) about the channel, or if we employ approximations of the truncated residuals to update the estimates at intermediate rate points prior to mode decision.

IV. Dual Frame Prediction and Pulsed-Quality Allocation

In dual frame prediction [14], two reference frames, one short and one long term, are used for motion compensation. The long-term frame is periodically updated every N frames. Later, in [11], pulsed quality (allocation of additional bit rate) was proposed for the long-term frames (while keeping the rest of the frames at a lower quality), leading to improved performance in error-prone scenarios.

Here, we investigate periodic long-term frames, both with even and with uneven (pulsed) quality. Pulsed-rate allocation takes place only at the base layer level. However, since we desire roughly equal-length base layers, we incur some extra delay for the pulsed frames, as shown in Fig. 1(b), where a delay of one frame is observed. The bitstream is displayed first on top

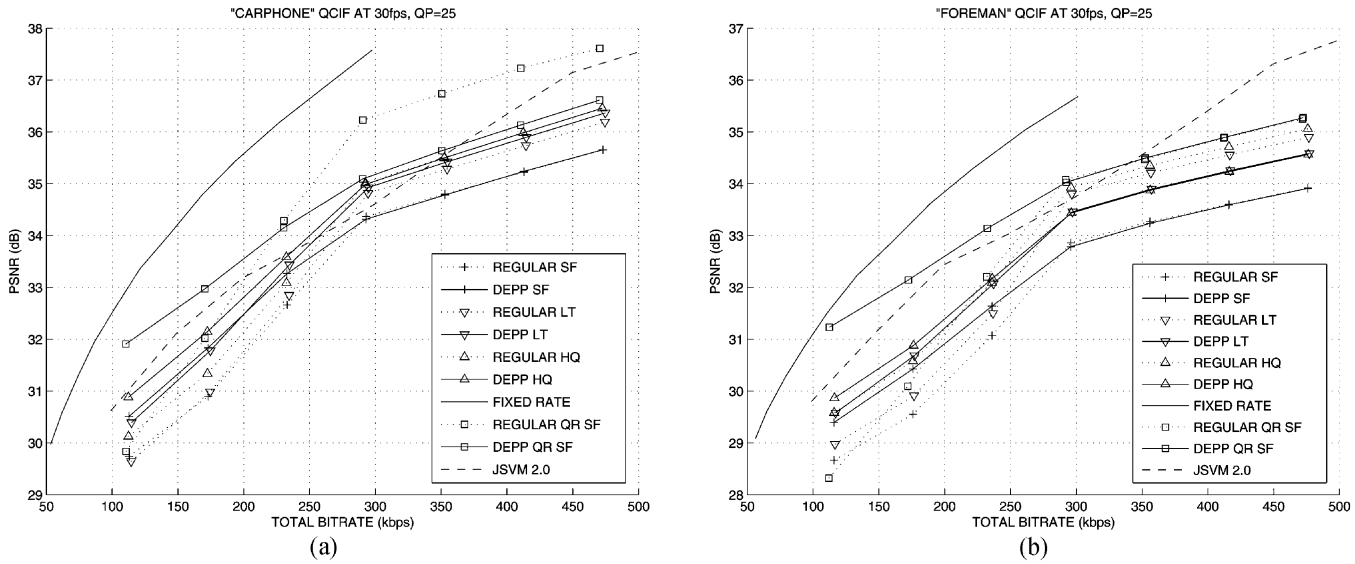

Fig. 2. Constant bit rate (CBR) truncation experimental PSNR performance versus total received bit rate for (a) "Carphone" at 30 fps and (b) "Foreman" at 30 fps.

as it is encoded and on the bottom as it is transmitted. Flattening the bandwidth and transmitting at constant rate ensures a constant-length base layer. The decoder receives this group of frames, extracts the overlayed rate belonging to the long-term frame, and then decodes them. Ensuring a constant and low average-rate base layer guarantees that it will not surpass the lowest rate threshold (imposed by the bottleneck channel; e.g, 64 kbps if the operational range includes ISDN). Otherwise, the rate pulses could surpass this threshold.

The encoder selects the reference frame and block through an exhaustive search whose goal is to minimize prediction distortion. We minimize the following prediction distortion measure from [6]:

$$\mathrm{SAD} = \mathrm{SAD}_b + \lambda_1 \mathrm{SAD}_{er} + \lambda_2 \|p_b - p_{er}\| \qquad (12)$$

where SAD_b is the prediction distortion from the base layer and SAD_{er} is the prediction distortion from the EL reference. The last term is identical to the one in (11) with the sole difference that $\|.\|$ denotes here SAD calculation. The λs are constants with values $\lambda_1 = 1.2$ and $\lambda_2 = 0.05$. Equation (12) is used both for block motion estimation as well as for reference frame selection. The rate-distortion constrained scheme of the H.264/AVC test model was not used. Minimizing just SAD_b would lead to suboptimal reference frame selection because we are not necessarily going to use LPLR mode for all macroblocks in the frame. The motion vectors (MVs), reference indices, and motion partitioning are encoded in the base layer and are re-used when coding the EL. The EL encodes the FGS residuals and the EL block coding mode.

V. Experimental Results and Discussion

We employed the H.26L-PFGS video codec, comprised of an H.264 TML9 base layer codec and an EL codec with MPEG-4 FGS syntax. A uniform quantization parameter (QP) value was applied to all blocks of the base layer: $\mathrm{QP} = 25$ for "Carphone" and "Foreman," $\mathrm{QP} = 27$ for "Container" and "Mother-Daughter." We measured the performance of the scalable codec by truncating the enhancement bit rate of each frame in 250 byte intervals (chunks). For sequences encoded at a frame rate of 10 fps, this translates to bitrate intervals of 20 kbps, while for sequences encoded at 30 fps this translates to 60 kbps. The bit rate horizontal axis in Figs. 2 and 4(b) corresponds to the total transmission bit rate, comprised of the base layer that naturally varies, but has been encoded so that it provides an acceptable visual quality [usually a peak signal-to-noise ratio (PSNR) value close to 30–31 dB], and the additional EL bit rate that comes in 250-byte chunks. The leftmost point in the curves of Fig. 2 corresponds to the base layer plus one 250-byte chunk.

Integer motion vectors are used for motion estimation and compensation. The loop filter is used but not modeled in our per-pixel estimates due to the high complexity. The use of integer MVs enabled optimal calculation of the estimates. Regarding efficient techniques for adapting per-pixel estimates to fractional pixel motion vectors, see [14], [15]. We set $f_0 = 1$ for $R_0 = 0.65 \times R_{ER}$, meaning that regardless of how many 20 kbps/60 kbps chunks of enhancement layer bits are received at the decoder side, the encoder runs its recursions by always assuming that network conditions force the enhancement layer to be truncated at some 65% of the rate needed for full reconstruction of the enhancement reference. The encoder is, thus, made to *assume* that there is drift on every enhancement reference, whether or not there actually is. Values greater than 0.65 would lower performance for low rates and raise it for higher rates.

All proposed schemes employ IPPP structure in both base and enhancement layer. The entropy coder was CABAC. We investigate both the scheme in [6] referred to as REGULAR, and our proposed scheme DEPP. The only difference between them is the modeling of drift. For each of the two schemes, three codec configurations are evaluated. The SF codecs employ single-frame prediction using the previous frame as the reference. The LT codecs employ periodic updating of an

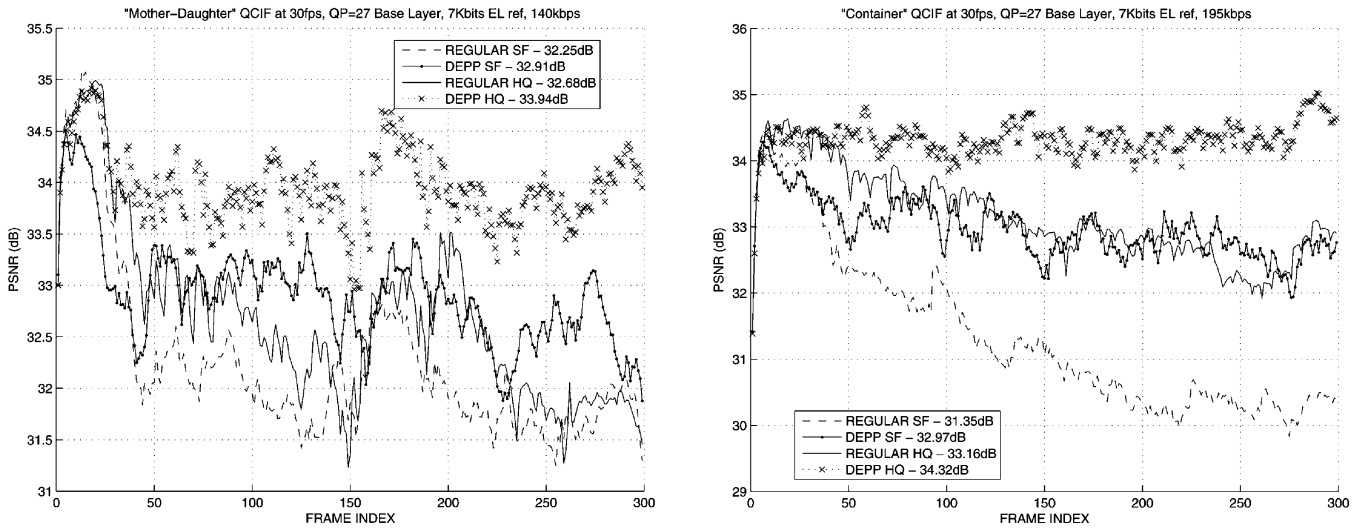

Fig. 3. Constant bit rate (CBR) truncation experimental PSNR performance versus frame number for (a) "Mother-Daughter" at 30 fps and (b) "Container" at 30 fps.

additional long-term frame buffer; hence, two reference frames are available during motion compensation. We recall that the reference frame is fixed for both layers and the decision is made at the *base layer* encoding step. Hence, a block in the enhancement layer will be predicted from the same (enhancement) frame as the base layer block was predicted from. No additional reference frame index is transmitted in the enhancement bitstream. Finally, the HQ codecs family employs pulsed quality on the long-term frame. The long-term frame is encoded with a finer quantization parameter QP_L than the rest of the frames which are instead coded with a coarser quantization parameter QP_S to ensure the same average bit rate as with SF and LT codecs. In our simulations, the updating period has been fixed to 5. The following QP combinations were used for each of the evaluated sequences: $(\mathrm{QP}_L, \mathrm{QP}_S) = (23, 26)$ for "Carphone" and "Foreman," $(\mathrm{QP}_L, \mathrm{QP}_S) = (23, 29)$ for "Container" and "Mother-Daughter." After searching over a range to determine a good value of the factor c in (11), c_{REGULAR} was fixed to 13 for static sequences (detected through motion vectors) and 4 for dynamic sequences. While optimizing the parameter individually for each sequence is not realistic, we consider that it is realistic that the encoder would be able to make this simple binary categorization to choose one of two values of the parameter. Then $c_{\mathrm{DEPP}} = 0.5 \times c_{\mathrm{REGULAR}}$ was used. The same value was used for SF, LT, and HQ versions of the codec.

Fig. 2 shows results for uniform truncation rate: The enhancement layers of every frame are truncated at the same bit length. In Fig. 2(a), all three curve families (SF, LT, and HQ), and SF in particular, show gains of 1 dB for DEPP at low to medium bitrates, compared to their respective REGULAR curves. The performance loss at high rates is negligible. A similar case is observed in Fig. 2(b), where this time the gains at low rates are smaller. REGULAR HQ and LT perform well at high rates hinting at the usefulness of multiple frame references for this sequence. DEPP again underperforms for high rates. Recall that c was optimized for SF codecs so our claims for LT and HQ are conservative and not representative of the maximum achievable performance. For reference we show the performance of the nonscalable SF codec ("FIXED RATE") with integer motion vectors. It is apparent that the generic FGS methodology achieves SNR scalability at a significant cost in compression efficiency.

In Fig. 2, we observe a "knee" in the curves where the slope changes significantly. This point corresponds to the EL reference truncation rate. It does not depend on the expected rate used by the drift estimation, which is why the knee occurs in both the REGULAR and DEPP curves. The reason for the knee is as follows: Up to the EL reference rate, having more rate for the EL helps improve both the prediction reference and the final display. If, however, the rate received is greater than the EL reference, the decoder will still only use the prescribed reference. So the extra rate is used only for final display purposes, but does not help with any prediction, which is why the slope is lower for that portion of the curves.

The Scalable Video Codec JSVM 2.0 [2] that incorporates FGS is also evaluated with IBPBPB structure (low delay) and integer motion vectors (performance suffers 1–1.5 dB compared to quarter-pixel vectors). It outperforms the older H.26L-PFGS codec as was expected, due to more advanced entropy coding and motion prediction. Last, we investigate performance when quarter-pel motion vectors are enabled while DEPP still models vectors as integers. The "DEPP QR" is now handicapped due to inaccurate modeling of the motion compensation process and this shows in Fig. 2(a). For Fig. 2(b), however, "DEPP QR" performs well compared to "REGULAR QR."

Next, in Fig. 3, we investigate performance for various truncated rates on a per frame basis. Due to space constraints we omit the LT codecs from this comparison. From both figures we observe that DEPP is always better than REGULAR, which was expected since the truncation rate was low to medium. However, we also observe the substantial gain through the use of pulsed quality (HQ). For sequences with repetitive image content, such as "Mother" and "Container," we observe gains of 1–1.5 dB. Note that pulsing the quality does not create artificially high

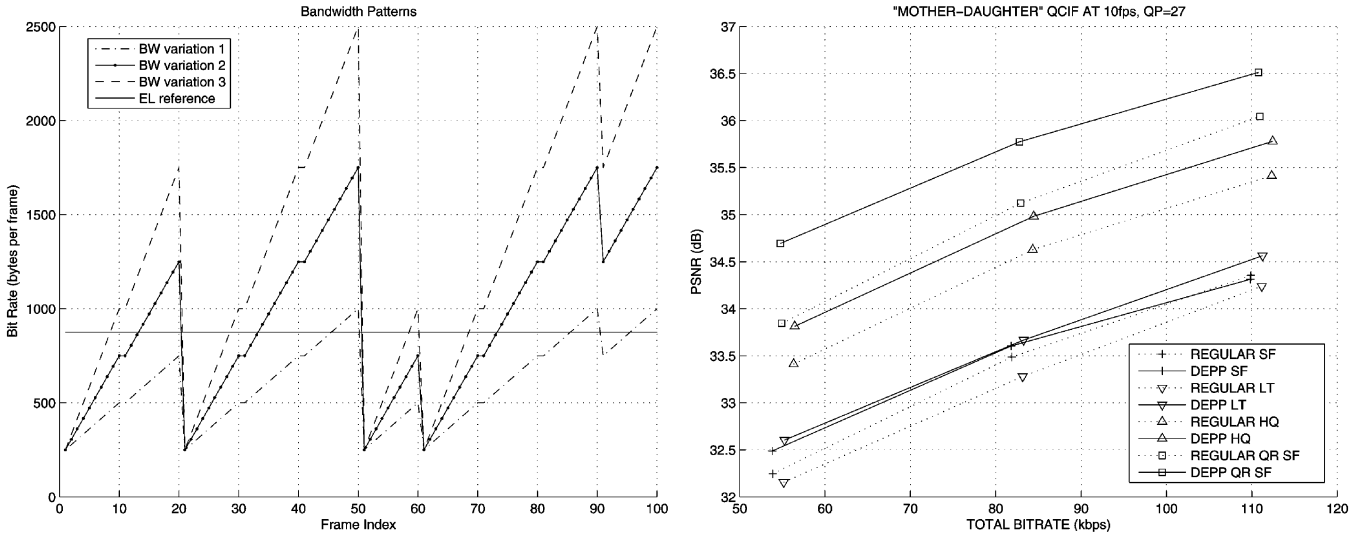

Fig. 4. Variable bit rate (VBR) truncation experiment. (a) Time-varying bit rate truncation pattern. (b) PSNR performance versus total bit rate received for "Mother-Daughter" at 10 fps.

variations in PSNR: similar PSNR spikes are found in the SF variants as well. Finally, we observe for the REGULAR codecs that their performance deteriorates with time steadily, in contrast to the DEPP codecs that are inherently resistant to drift. The PSNR values inside the legend boxes are the averaged values over the entire sequence.

Finally, in Fig. 4(b), we investigate variable bandwidth scenarios. The left, center, and right points in the curves in Fig. 4(b) correspond to the bit rate truncation patterns 1, 2, and 3 in Fig. 4(a), respectively. The EL reference truncation rate is depicted with a straight line. The shape of the time-varying truncation rate patterns was chosen to resemble TCP/IP behavior. Fig. 4(b) shows that DEPP performs well, though the margin against REGULAR is not as high as previously. DEPP LT is not noticeably better than DEPP SF. The reason is the low quality long-term reference base layer, whose SAD contributes to reference frame and MV selection in (12). Furthermore, the low-quality BL makes the evaluation of fractional pixel displacements [16]—a primary reason for the compression efficiency of multiple frame prediction—hard. Once it is pulsed, we observe impressive gains in the HQ codecs. Last, the "QR" curves use quarter-pixel MVs while the recursive estimates model them as integer only. DEPP outperforms REGULAR, though with a smaller margin.

The additional computations consist of two parts: FGS decoding (inverse DCT and inverse quantization) that yields the intermediate decoded residual, and the recursive updating step for each of the moments once the EL bitstream has been fully produced. The complexity of FGS decoding is very close to that of FGS encoding since the operations are simply reversed. The complexity of the updating step is essentially equal to the complexity of the algorithm in [13], which is comparable to applying DCT and quantization. As we track two moments, the updating complexity is estimated to be twice the decoding complexity. The overall complexity of our scheme is, thus, approximately three times the decoding complexity. We found that execution time is increased by just 3% when DEPP is employed.

VI. Conclusion

The proposed drift estimation approach yielded performance gains of about 1 dB for most sequences across low to medium rates, with negligible loss at high rates. This was true, even though the encoder persisted with a simplistic assumption about the truncation rates, an assumption that did not hold true in the actual simulations, for which the enhancement reference truncation rates varied substantially. The reason is that, even for a crude channel description, it is better to assume some amount of drift and estimate its effect rather than disregarding it altogether. Pulsed-quality long-term frame prediction was shown to be advantageous for low-to-medium rates and video content with sufficient temporal redundancy.

Future work can include modeling drift in the evolving SVC standard [2]. FGS is used in an LPLR coding approach that encodes base layer motion-compensated residuals to achieve SNR scalability. Prediction from EL frames, similarly to HPLR and HPHR coding modes, can be used to improve the compression efficiency of the FGS layer, introducing potential drift.

References

[1] A. Leontaris and P. C. Cosman, "Optimal per-pixel estimation for scalable video coding," in *Proc. IEEE Int. Conf. Image Processing*, Oct. 2004, pp. 2087–2090.

[2] J. Reichel and H. S. Wien, Scalable Video Coding Working Draft 1 2005, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-N020.

[3] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 301–317, Mar. 2001.

[4] F. Wu, S. Li, and Y.-Q. Zhang, "A framework for efficient progressive fine granularity scalable video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 332–344, Mar. 2001.

[5] X. Sun, F. Wu, S. Li, W. Gao, and Y.-Q. Zhang, "Macroblock-based progressive fine granularity scalable video coding," in *Proc. IEEE Int. Conf. Multimedia Expo.*, 2001, pp. 461–464.

[6] Y. He, F. Wu, S. Li, Y. Zhong, and S. Yang, "H.26L-based fine granularity scalable video coding," in *Proc. IEEE ISCAS*, 2002, vol. IV, pp. 548–551.

[7] F. Wu, S. Li, B. Zeng, and Y.-Q. Zhang, "Drifting reduction in progressive fine granularity scalable video coding," presented at the Int. Picture Coding Symp. Apr. 2001.

[8] C. Zhu, Y. Gao, and L.-P. Chau, "Reducing drift for FGS coding based on multiframe motion compensation," in *Proc. IEEE ICASSP*, May 2004, vol. 3, pp. 253–256.
[9] Y. Zhou, X. Sun, F. Wu, H. Bao, and S. Li, "Flexible P-picture (FLEXP) coding for the efficient fine-granular scalability (FGS)," in *Proc. IEEE Int. Conf. Image Processing*, Oct. 2004, vol. 3, pp. 2071–2074.
[10] J. Ascenso and F. Pereira, "Drift reduction for a H.264-AVC fine grain scalability with motion compensation architecture," in *Proc. IEEE Int. Conf. Image Processing*, Oct. 2004, vol. 4, pp. 2259–2262.
[11] A. Leontaris, V. Chellappa, and P. C. Cosman, "Optimal mode selection for a pulsed-quality dual frame video coder," *IEEE Signal Process. Lett.*, vol. 11, no. 12, pp. 952–955, Dec. 2004.
[12] S. Han and B. Girod, "Robust and efficient scalable video coding with leaky prediction," in *Proc. IEEE Int. Conf. Image Processing*, Sep. 2002, vol. 2, pp. 41–44.
[13] R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 966–976, Jun. 2000.
[14] A. Leontaris and P. C. Cosman, "Video compression with intra/inter mode switching and a dual frame buffer," in *Proc. IEEE DCC*, Mar. 2003, pp. 63–72.
[15] V. Bocca, M. Fumagalli, R. Lancini, and S. Tubaro, "Accurate estimate of the decoded video quality: Extension of ROPE algorithm to half-pixel precision," presented at the Int. Picture Coding Symp. Dec. 2004.
[16] A. Chang, O. C. Au, and Y. M. Yeung, "A novel approach to fast multi-frame selection for H.264 video coding," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, May 2003, vol. 3, pp. 413–416.

Athanasios Leontaris (S'97–M'06) 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.

He was a summer intern at AT&T Labs—Research, New Jersey, and at NTT Network Innovation Labs, Japan, in 2004 and 2005, respectively. His research interests include image and video compression, video transmission, multimedia processing, and image quality modeling.

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 a Professor and Director of the Center for Wireless Communications. 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 (1998 to 2001), a Guest Editor of the June 2000 special issue of the IEEE JOURNAL ON SELECTED AREAS IN COMMUICATIONS (JSAC) on "error-resilient image and video coding," and the Technical Program Chair of the 1998 Information Theory Workshop, San Diego. She was an Associate Editor of the IEEE SIGNAL PROCESSING LETTERS (2002 to 2005). She was a Senior Editor (2003 to 2005) and is currently the Editor-in-Chief of the IEEE JSAC. Her Web page address is http://www.code.ucsd.edu/cosman/.