END-TO-END DELAY FOR HIERARCHICAL B-PICTURES AND PULSED QUALITY DUAL FRAME VIDEO CODERS

Athanasios Leontaris[†] *and Pamela C. Cosman*[*]

[†]Dolby Laboratories, Inc., Burbank, CA 91505, U.S.A.
[*]University of California at San Diego, La Jolla, CA 92093, U.S.A.

ABSTRACT

Real-time video applications require tight bounds on end-to-end delay. Hierarchical bi-directional prediction requires buffering frames in the encoder input buffer, thereby contributing to encoder input delay. Long-term frame prediction with pulsed quality requires buffering at the encoder output, increasing the output buffer delay. We compare the end-to-end delay of these two approaches using simulations to determine the delay vs. compression efficiency trade-off.

Index Terms— H.264, hierarchical B-pictures, long-term frames.

1. INTRODUCTION

Constraining delay is important for real-time communication and live event broadcast. Live television broadcast should have a delay of no more than 1 second in many cases. Interactive video-phone communication should have a maximum end-to-end delay of no more than 300ms. For traditional predictive coding techniques (the IPPP coding structure), the end-to-end delay is low. A frame is captured, encoded in real-time, briefly buffered, and then transmitted. After brief buffering, the decoder decodes the bits and displays.

By filtering across frames or by using bidirectional prediction, compression performance improves because the temporal correlation among several neighboring frames is better exploited, but additional delay is incurred. An example is motion-compensated temporal filtering (MCTF). Trade-offs of delay and compression in MCTF video codecs were investigated in [1]. In that work, delay was reduced by selectively removing the update step. Recently, the update step was removed from the working draft of the Scalable Video Coding extension to H.264/AVC [2]. The end-to-end delay trade-off for MCTF was studied in [3]. Delay is an issue for hierarchical bi-predictive structures as well. The delay in the hierarchical case depends on the size of the Group of Pictures (GOP), and cannot be reduced by removing update steps while keeping GOP size intact.

One can also have increased delay when using a single-direction (forward) prediction scheme. The codec proposed in [4] employs two reference frames, one short-term (ST) and one long-term (LT). The LT frame is afforded extra bits; it is high (pulsed) quality. At a given constant transmission bit rate, these frames will take longer to transmit, introducing delay. The rest of the frames are starved to achieve the rate constraint. Compression efficiency was improved for certain image sequences but delay was not studied in that work.

The studies in [1, 3] did not take into account the effect of the encoder output and the decoder input buffering requirements which are non-trivial. We model both delays. In this work, we study the delay for LT pictures with pulsed quality, as well as for hierarchical B-frames for varying GOP size. MCTF structures are not evaluated as they were found to be in most cases inferior to hierarchical B-frames [5]. Rate-control is used in all codecs to ensure that the delay budget is enforced: no buffer overflows occur.

The paper is organized as follows: In Section 2 we define and discuss the end-to-end delay in detail. The investigated video coders along with the rate control schemes we adopted are discussed in Section 3. Experimental results and conclusions are in Section 4.

This work was performed while the first author was at UCSD. It was supported in part by the Center for Wireless Communications at UCSD, the Office of Naval Research, and the UC Discovery Grant program of the State of California.

2. END-TO-END DELAY

End-to-end delay involves delay at the source encoder, channel encoder, channel decoder, and source decoder, as well as transmission and propagation delay. We assume a propagation delay of zero, and we assume a lossless channel, so we do not include channel coding. We further ignore computation time at the encoder and decoder, limiting our scope to the buffers at the source encoder, shown in Fig. 1, the transmission delay, and the buffers at the source decoder.

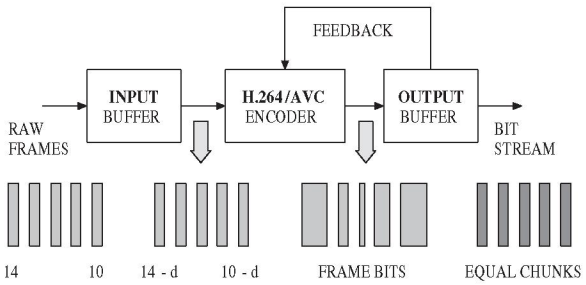


Fig. 1. The encoder input buffer and output buffer introduce delay.

The encoder converts the frame into a bit stream instantaneously and then starts writing the bits to the encoder output buffer at a constant rate. If frame i is encoded with b_i bits, then it is written into the buffer at a rate of $30b_i$ bits/sec, as the video is at 30 frames per second, so the bits for each frame get written during 1/30th of a second. The rate $30b_i$ may be more or less than the average source coding rate r. The encoder output buffer is a "leaky bucket": it is continuously drained at the constant average source coding bit rate r. Additional delay may be added by buffering the input frames prior to feeding them to the encoder. The encoder input buffer delay depends on the motion-compensation structure used, and varies in increments of whole frame durations.

The encoder output buffer determines how tightly the rate control must operate. With a constant source coding rate of r bits per second, each frame could have the same exact $r/30$ bits per frame (for 30 frames per second one frame is displayed for 33ms), and then the output buffer could be in fact of zero length. Bits leave the buffer as soon as they enter it. Still, even in that case, 33ms are required for a frame to leave the encoder and arrive at the decoder in its entirety. This is termed the transmission delay D_{TX}. But then the encoder could not respond to a scene cut or to high motion by using more bits, and by giving fewer bits to static scenes. Allowing the encoder output buffer to be larger leads to higher video quality.

We will require our rate control to live within the buffer without any frame skipping and with producing no overflows or underflows. During the time (33ms) that the bits for frame i are fed into the output buffer, $r/30$ bits will drain out of it. Therefore the rate control must ensure that the length in bits of any single frame is no longer than the buffer size plus $r/30$ bits, and indeed, is no longer than the space remaining in the buffer at that particular time plus $r/30$ bits.

The decoder is a mirror image of Fig. 1. Bits are buffered in a decoder input buffer, which is the same size as the encoder output buffer, a common assumption made in [6]. Decoded frames are buffered at the output prior to display. In our model which excludes delays from computation, the source coding end-to-end delay D_{e2e} depends on the four buffers and can be written as:

$$D_{e2e} = D^{in}_{enc} + D^{out}_{enc} + D_{TX} + D^{in}_{dec} + D^{out}_{dec} \qquad (1)$$

where subscripts indicate encoder or decoder, and superscripts indicate input or output buffers.

With hierarchical B-frames, the encoder cannot begin to encode a frame until the entire GOP is available for processing, so for GOP size equal to N_{GOP} the encoder begins processing one frame while $N_{GOP} - 1$ frames are in the encoder input buffer. Thus $D^{in}_{enc} = (N_{GOP} - 1)t_{fr}$, where t_{fr} is the display time duration of a frame. In all our experiments we encode at 30 frames per second, so $t_{fr} =$ 33ms. The transmission time is $D_{TX} = t_{fr}$. We assume $D^{out}_{enc} = D^{in}_{dec}$. Finally, the output/display decoder delay is again: $D^{out}_{dec} = (N_{GOP} - 1)t_{fr}$. We thus rewrite Eq. 1 as:

$$\begin{aligned} D_{e2e} &= (N_{GOP} - 1)t_{fr} + 2 \times D^{out}_{enc} + t_{fr} + (N_{GOP} - 1)t_{fr} \\ &= 2 \times D^{out}_{enc} + (2 \times N_{GOP} - 1)t_{fr} \end{aligned} \qquad (2)$$

If the rate control could assign exactly $r/30$ bits per frame, then no buffering would be needed at the encoder output or decoder input, and the above result shows that the delays for N_{GOP} equal to 1, 2, 4 would be 1, 3, and 7, respectively, times the frame duration of 33ms.

3. COMPARED ENCODERS

We evaluate three types of encoders: predictive IPPP coding (SF), long-term prediction with pulsed quality (LT-HQ), and hierarchical B-frames (GOP). The codecs are now described in detail.

The *SF* codec, shown in Fig. 2, is based on the Joint Model (JM) 10.1 reference software of the H.264/AVC video coding standard [7]. Frames are encoded predictively in an IPPP structure. Two short-term reference frames were used for motion compensation. From Eq. 2 the end-to-end delay is $D_{e2e} = 2D^{out}_{enc} + t_{fr}$. Note that both SF and LT-HQ have $N_{GOP} = 1$. We used the rate control algorithm included in the JM 10.1 reference software and described in [8].

The *LT-HQ* codec, shown in Fig. 2, uses a short-term (ST) reference frame and a long-term (LT) reference frame for motion compensated prediction as described in [4]. It is based on a modified version of the JM 10.1 reference software. The LT reference frame

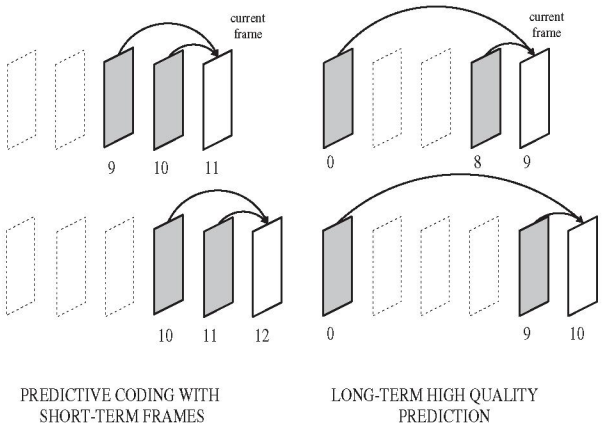


Fig. 2. The LT-HQ and the SF motion-compensated structures. The arrows denote motion-compensated prediction.

is periodically updated every U frames and is afforded more bits than the regular frames. We chose $U = 5$, and allocated two or three times as many bits to the LT frames as to the regular frames. The exact number of bits is calculated adaptively so that the overall rate constraint is satisfied. The decisions to allocate twice or thrice the short-term bits and to set $U = 5$ are not optimal. Better performance could be achieved by optimizing these parameters, but exploring this large parameter space is beyond the scope of this paper. As in the SF case, the end-to-end delay is $D_{e2e} = 2D^{out}_{enc} + t_{fr}$, but now the output buffer will tend to be larger for good performance, since the high quality frames require more bits.

In LT-HQ the rate allocation is similar to that in [8] with some critical modifications. We do not allocate rate to ST and LT frames from a common budget. The budget is divided into two bins: ST and LT rate bins. Two separate rate control "paths" for ST and LT frames draw bits from the respective bins. They however share the constraint on the encoder buffer status to avoid a buffer overflow. In each rate control path, the buffer limit is enforced by modifying the QP to achieve the target rate but also by the last-resort measure of forcing SKIP coding modes on blocks when the buffer is about to overflow. We switch the QP on a basis (*basic unit*) of 11 macroblocks (MB). The quadratic model of [8] selects a QP for this *basic unit* that *ought* to avoid an overflow. Since the quadratic model is only an estimate, there are many cases where SKIP modes must be invoked. Signaling a SKIP mode for a MB involves transmitting two bits. In that case, the reconstructed MB is a motion-compensated prediction from a previous reference frame. The motion vector is obtained through spatial prediction of neighboring motion vectors.

The third coder we studied uses hierarchical motion-compensated prediction, shown in Fig. 3. In hierarchical structures B-frames can be predicted from other B-frames. In non-hierarchical structures they are predicted from the closest P or I-frames. This coder was implemented with the JSVM 3.3.1 reference software [2]. We constrained the codec so that the generated bitstream is fully H.264/AVC compliant. For $N_{GOP} = 2$ we obtain the well-known IBPBPB prediction structure where one B-frame is encoded between two P or, alternatively, I-frames. Hierarchical structures benefit from prediction both from the "future" and the "past". It is particularly advantageous in cases of global motion and camera pan as shown in [9]. Note that the "closed loop" approach [5] was used for hierarchical

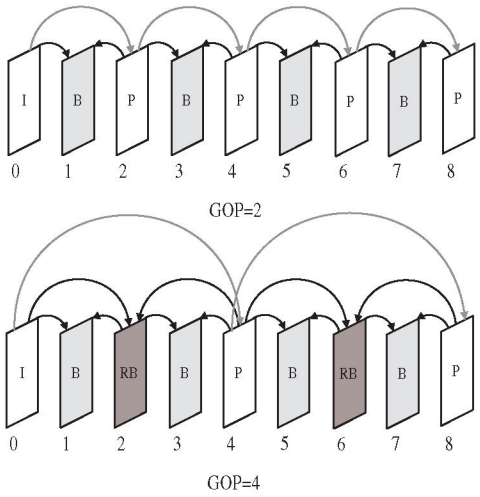


Fig. 3. Hierarchical bi-predictive motion-compensated structures. The arrows denote motion-compensated prediction. The RB frames are B-frames that can be used as *references*.

prediction: B-frames are predicted from the reconstructed reference frames and not the original ones as traditionally done in MCTF.

The JSVM 3.3.1 reference software does not include rate control for the base layer. As a result, we adopted the rate control scheme from the JM software [8]. This rate control scheme was not designed with hierarchical motion-compensation structures in mind. It addresses B-frames in non-hierarchical structures.

While the rate of the P-frames is strictly controlled by changing the QP in terms of basic units, the B-frames are allocated a single QP value for the entire frame. Thus, the rate is not explicitly controlled. In general, for constant QP allocation, a B-frame will be noticeably smaller than its neighboring P-frames, due to the efficiency of bi-directional prediction. Furthermore, the rate control of [8] allocates a QP incremented by two over the average QP of the neighboring P-frames. The B-frame is thus guaranteed to be smaller than the neighboring P-frames. We modified it further for the hierarchical GOP-based encoder:
(a) When the number of remaining blocks times the two bits for signaling SKIP mode are close to triggering a buffer overflow, the rate expended by the B-frames is controlled by selecting during rate-distortion optimization the mode that uses the fewest bits.
(b) The B-frames in the first hierarchical level (the ones predicted directly from neighboring P or I-frames) are allocated a QP value that is the average of the reference frame QPs incremented by two.
(c) The B-frames in the higher hierarchical levels (e.g. frames 1 and 3 of Fig. 3) are allocated a QP value that is incremented by two compared to the QP value used by the previous hierarchical level.

We note that large values of N_{GOP} such as 8 and 16 are possible (the H.264/AVC specification allows up to 16), but preliminary trials showed that the gain in PSNR is small compared to the dramatic increase in end-to-end delay. For SF and LT-HQ, the output frame order is [012345678]. For the $N_{GOP} = 2$ case, the order is [021436587]. For $N_{GOP} = 4$, the order is [042138657].

All investigated video codecs are fully compatible with H.264/AVC [7]. The results in Section 4 were obtained with the JM 10.1 reference decoder for all bit streams. Although we ignored the encoder complexity during delay calculation, we constrained it to be approximately equal for all four types of streams. The hierarchical codecs use one short-term reference frame for the P-frames. However, the B-frames are encoded with bi-directional prediction, roughly equal to prediction from two frames. Additional iterations, needed for bi-prediction to converge, may lead to greater complexity. The LT-HQ codec uses two reference frames. We hence used two short-term reference frames for the SF codec as well.

4. RESULTS

We investigated the performance of the four codecs for a variety of video sequences: *Carphone*, *Mobile-Calendar*, and *Flower-Garden*. In Figs. 4 and 5 we show video quality versus end-to-end delay. The bit rate is fixed for all curves displayed within the graphs. The delay was varied by allocating different numbers of bits to the encoder output buffer (D_{enc}^{out}). Performance increases with delay and GOP size. $N_{GOP} = 4$ outperforms $N_{GOP} = 2$, which in turn outperforms $N_{GOP} = 1$, both SF and LT-HQ. LT-HQ is better than SF for low bit rates. For high rates, the importance of a high-quality LT frame naturally diminishes, and, with it, the performance advantage becomes negligible.

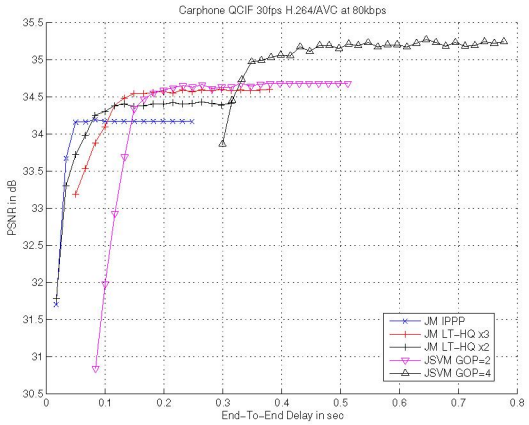
We observe in Figs. 4 and 5 that the SF codec achieves good performance at a delay of around 51ms. The minimum delay t_{fr} is 33ms, so adding two times an encoder output buffer delay slightly higher than 9ms should give good performance. Since the rate control's QP basic unit is set to 11, it proves stable, and increasing the delay has no effect on the performance. If, however, the basic unit is set to, say an entire frame, then delay increases significantly, as does the performance. However, investigating this trade-off would yield a very large parameter space. The basic unit was thus fixed to 11.

The LT-HQ codec achieves good performance at a delay of around 77ms for $2\times$ pulsing. We note that the minimum delay that guarantees good performance can be calculated from U and the long-term to short-term bit budget ratio. The PSNR performance depends on the image sequence. For the highly active "Flower" there is no gain over the SF codec. Significant gains are however observed for the "Carphone" and "Mobile" sequences. For $3\times$ pulsing, both the delay, at 117ms, as well as the performance is slightly higher.

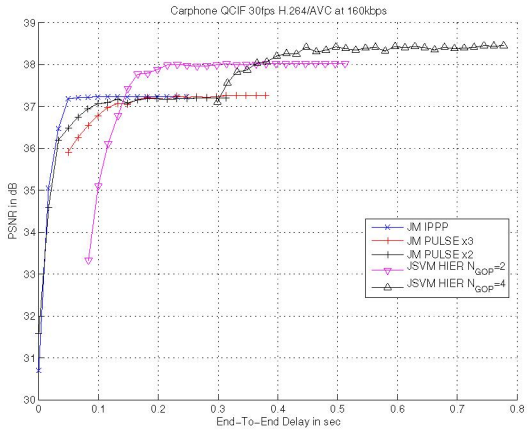
Moving to the $GOP = 2$ case, we observe that the end-to-end delay needs to be at least 170ms for good performance. There is no performance gain over LT-HQ for "Carphone". However, impressive gains are observed in "Mobile" and "Flower". It is thus evident that hierarchical structures can be very advantageous in static sequences or sequences with global motion.

The increase of the GOP size to 4 increases delay considerably to more than 300ms. Apart from increased GOP delay, the anchor P-frames get large contributing further to delay. The three B-frames in each GOP need many fewer bits to be encoded. In terms of performance gain, "Carphone" and "Mobile" benefit the most. The largest gain in "Mobile" is attributed not only to its global motion but also to the fact that it is translational.

Conclusions: We studied end-to-end delay versus compression performance when employing video encoders with varying GOP size. In addition, we investigated the effect on delay of giving some frames significantly more bits than the rest. All these codecs are H.264/AVC compliant. We also implemented a rate control algorithm for the LT-HQ codec. The work in [4] used constant QPs without any consideration of rate or delay constraints. Here we operated under both constraints, including rate control implemented for the hierarchical B-frames. Compared to previous work, we took into account the encoder output and the decoder input buffer delays.

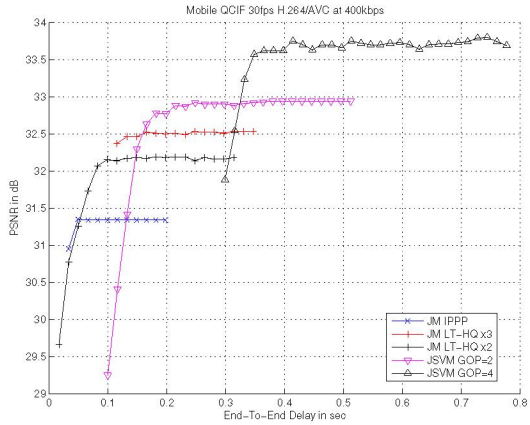


(a)

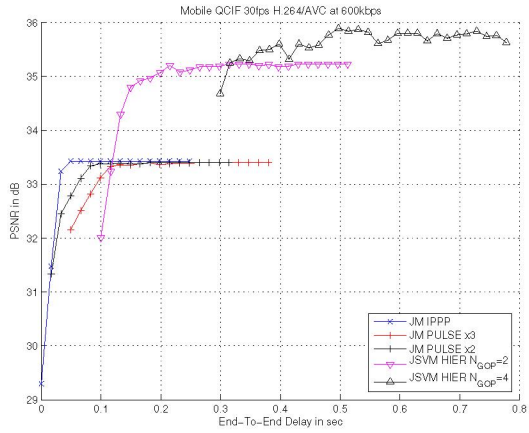


(b)

Fig. 4. "Carphone" QCIF 176 × 144 at 30fps PSNR vs. delay (in seconds) for fixed source coding bit rate. (a) 80 kbps. (b) 160 kbps.



(a)



(b)

Fig. 5. "Mobile" QCIF 176 × 144 at 30fps PSNR vs. delay (in seconds) for fixed source coding bit rate. (a) 400 kbps. (b) 600 kbps.

Let us summarize our conclusions from the codec evaluation: (a) LT-HQ is advantageous for relatively static sequences with repetitive content. (b) $N_{GOP} > 1$ structures benefit from static sequences and from sequences with global motion. (c) As N_{GOP} increases, the gain is non-trivial only if the sequence is either quite static, or if the global motion is translational. (d) The delay thresholds are as follows: between 51ms and 77ms SF is the best choice, between 77ms and 170ms LT-HQ performs well, the large space between 170ms and 300ms is dominated by $N_{GOP} = 2$, and for delays larger than 300ms then $N_{GOP} = 4$ is the best choice. Delays larger than 300ms are prohibitive for real-time communication, however.

5. REFERENCES

[1] G. Pau, B. Pesquet-Popescu, M. van der Schaar, and J. Vieron, "Delay-performance trade-offs in motion-ccompensated scalable subband video compression," in *Proc. Advanced Concepts for Intelligent Vision Systems*, Sept. 2004.

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

[3] G. Pau, J. Vieron, and B. Pesquet-Popescu, "Video coding with flexible MCTF structures for low end-to-end delay," in *Proc. IEEE Int. Conf. on Image Proc.*, Sept. 2005.

[4] A. Leontaris, V. Chellappa, and P. C. Cosman, "Optimal mode selection for a pulsed-quality dual frame video coder," *IEEE Sig. Proc. Lett.*, vol. 11, no. 12, pp. 952–955, Dec. 2004.

[5] H. Schwarz, D. Marpe, and T. Wiegand, "Comparison of MCTF and closed-loop hierarchical B pictures," Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, JVT-P059, July 2005.

[6] M. Isnardi, "MPEG-2 video compression," SMPTE Tutorial Overview, Sarnoff Corporation, Nov. 1999.

[7] T. Wiegand, "Final draft international standard for joint video specification H.264," Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, JVT-G050, Mar. 2003.

[8] K.-P. Lim, G. Sullivan, and T. Wiegand, "Text description of joint model reference encoding methods and decoding concealment methods," Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, JVT-K049, Mar. 2004.

[9] M. Karczewicz and Y. Bao, "Need for further AVC test model enhancements," Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, JVT-L034, July 2004.