**Title**
Dual Frame Video Coding with Pulsed Quality and a Lookahead Window

**Authors**
Tiwari, M
Cosman, P

**Publication Date**
2006-03-01

# Dual Frame Video Coding with Pulsed Quality and a Lookahead Window

Mayank Tiwari and Pamela Cosman
Department of Electrical and Computer Engineering,
9500 Gilman Drive, University of California, San Diego, La Jolla, CA - 92093-0407
Email: {mayank,pcosman}@ece.ucsd.edu

### Abstract

In dual frame video coding, one short-term reference frame and one long-term reference frame are available for motion compensation. In prior research, it was shown that overall video quality was improved by allocating bits unevenly among frames so as to periodically create a high-quality frame that could serve as the long-term reference frame for some time. We extend this work to a cognitive radio scenario where pulses of extra bandwidth can be rented by the second, but rental agreements can be cancelled on short notice if the legacy user of that spectrum returns. With a scalable video coder, this pulse of extra bandwidth can be used to improve the quality of the current frame being encoded, or of the past long term frame, or can be spent on future frames if the encoder has access to them in advance. We compare these various uses to explore the advantages of allocating some of the available bandwidth to past or future frames.

## 1 Introduction and Background

Contemporary hybrid video codecs use motion-compensated prediction to efficiently encode a raw input video stream. For each block in the current frame to be encoded, the encoder searches in the reference frame (usually the immediate past frame) to find the best match block for it. The best match block is often called the prediction of the current block. The difference between the current block and its prediction from the reference frame is compressed and transmitted, along with the displacement (motion) vector that describes the location of the best match block relative to the current block. Called *inter* coding, this is the basic approach found in the video coding standards MPEG, MPEG-2, MPEG-4 [1], H.263 [2] and H.264/AVC [3].

In multiple frame prediction, more than one past frame is used in the search for the best match block. At the cost of extra memory storage and extra complexity for searching, multiple frame prediction has been shown to provide a clear advantage in compression performance. Papers on multiple reference frame motion compensation include [4, 5, 6, 7, 8, 9, 10, 11, 12, 13].

In dual frame video coding [7, 10, 11, 12, 13], two frames are used for inter prediction, a short-term reference (STR) and a long-term reference (LTR), as shown in Figure 1. Both encoder and decoder store LTR and STR frames. For encoding frame $n$, the STR is frame $n-1$ and the LTR frame is frame $n-k$, for some $k>1$. The LTR frame can be chosen by jump updating [8, 9], in which the LTR frame remains the same for encoding $N$ frames, then jumps forward by $N$ frames and again remains

Proceedings of the Data Compression Conference (DCC'06)
0-7695-2545-8 /06 $20.00

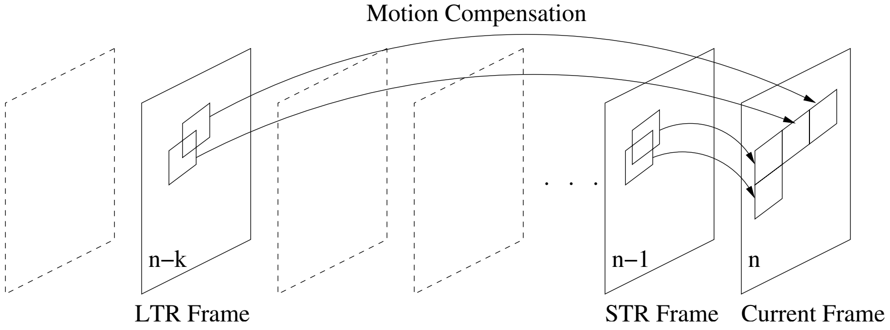


Figure 1: Dual frame video coding

the same for encoding the next $N$ frames. In continuous updating, every frame has a turn serving as an STR and as an LTR. In jump updating, every frame serves as an STR, but only every $N^{th}$ frame serves as an LTR; this allows the use of high quality LTRs, where the LTR frames are allocated more bits than regular frames. This has been shown to enhance the quality of the entire sequence [11]. In [11], the assumption was that certain frames could be starved of bits, and others given more bits, provided that a long-term average bit rate constraint was met.

In this paper, we are concerned with a somewhat different scenario: video transmission where we have a constant low bandwidth channel that is occasionally supplemented by the rental of a substantially larger bandwidth for a short interval of time. This type of scenario could occur with a cognitive radio, a radio that is capable of changing transmission or reception parameters on its own to complete a task [14, 15, 16]. One of the primary goals of a cognitive radio is to enhance the spectral efficiency of allocated spectral bands by opportunistically making use of regions in the spectrum that are temporarily free of traffic, utilizing orthogonal dimensions such as time, frequency, and space. This is accomplished by sensing the channel, and then adapting key parameters of the transmit waveform such as modulation format, power, bandwidth, frequency location, and code rate. Much existing wireless spectrum is under-used [17], providing an opportunity for a cognitive radio to rent extra bandwidth by shifting to a different spectrum for as little as one second or as long as a few hours. As envisioned in [15], when the spectrum to be rented has a legacy user, such as a fire department, the user who rents it might have to stop transmitting after every 20 millisecond interval in order to listen for 5 milliseconds to see if the legacy user has come back on the air. If he has, then the rental agreement is cancelled. In this way, spectrum could be used more efficiently, while providing minimal inconvenience to the primary user in the form of a small start-up delay.

We are concerned specifically with a situation where a video transmitter is able to rent extra bandwidth for short periods of time. The rental might be as short as 1 second. However, it might last for less time than that, since the rental agreement is cancelled if the legacy user comes on. We here consider the case where the extra rented bandwidth lasts only 30 milliseconds, or approximately the time duration of one frame at 30 frames per second. We will use dual frame video coding to exploit this extra bandwidth and enhance the overall video quality.

This paper is an extension to the work presented in [12]. In that work, extra bandwidth was deployed solely to encode the current frame at high quality. In the extension presented here, we use a scalable video coder, so the extra bandwidth can be used to enhance past frames or, in the case of a lookahead buffer, future frames.

This paper is organized as follows. Section 2 describes the scenario for extra bandwidth when there is no lookahead buffer, and the high bandwidth pulse can only be distributed over current and past frames. Section 3 presents results for the case where the encoder has a lookahead buffer, and the pulse can be distributed over future frames as well. Conclusions and future work appear in Section 4.

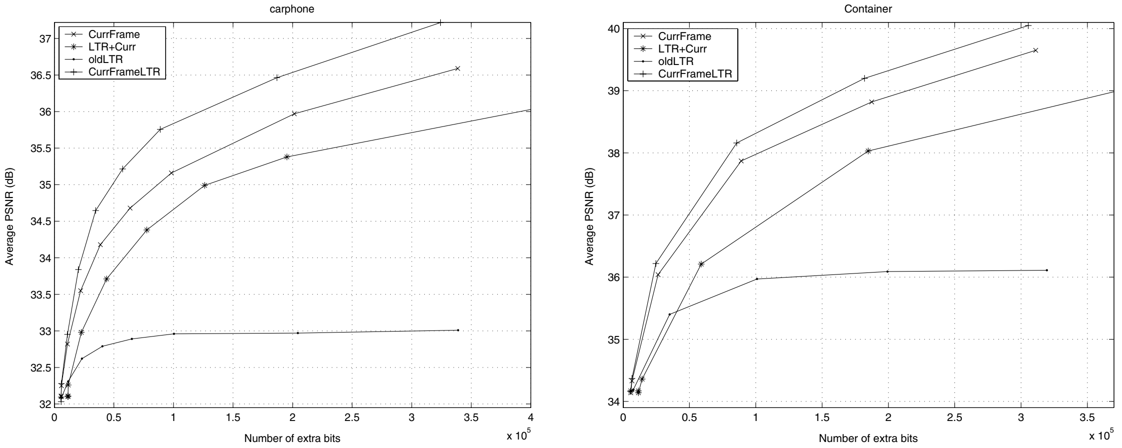
## 2 Scenario with No Delay

We first consider a real-time scenario with zero input buffering delay. By this we mean that the encoder takes in M frames per second, and each frame must be compressed within $1/M^{th}$ of a second, at the time when it arrives. No future frames are available. Therefore, if a high bandwidth pulse of spectrum is available in the current time slot, for a single reference frame encoder, the high bandwidth pulse must be used on the current frame, because past frames have already been encoded (and for a single reference frame encoder, there can be no possible advantage in enhancing the quality of past frames which have already been decoded and displayed), and future frames are not yet available.

For a dual frame encoder, however, there are more choices on deploying the pulse of extra bandwidth. We can allocate all bits to the current frame (we refer to this strategy as **CurrFrame**). We can allocate all bits to the current frame, and make it the new LTR frame for the next N frames, unless another pulse of extra bandwidth comes along in that time (**CurrFrameLTR**). We can enhance the quality of whichever *past* frame is currently being used as the LTR frame by allocating all bits to it (**oldLTR**). Unlike the case of the single reference frame encoder, in the dual frame case, there might be an advantage to spending extra bits on enhancing a past frame, even if those enhancement bits will not arrive at the decoder in time for the display of that frame, and will therefore not contribute to increasing the PSNR of that frame. Because that frame is the long-term reference, the enhancement can serve to improve the PSNR of future frames which refer back to it. Lastly, we can consider allocating some bits to the LTR frame and the remainder to the current frame (**LTR+Curr**). We note that the last two approaches listed are not standard compatible.

**Experimental Setup:** Simulations were performed using the H.264/AVC video codec [18] baseline profile which supports entropy coding with context-adaptive variable-length codes (CAVLC). We modified the H.264 reference software JM 9.6, obtained from [19]. The STR and LTR reference frames were stored in *list0*.

We used the 4:2:0 QCIF (176 × 144 pixels) video sequences Foreman, Carphone, Container, Mother-daughter, and Claire at 15 fps. We used 9 slices per frame, with 11 MBs per slice. We encoded 20 frames with the first one intra coded and the rest inter coded. A lossless channel was assumed with a constant low-rate bandwidth of



(a) Carphone at baseline 31 kbps

(b) Container at baseline 30 kbps

Figure 2: PSNR of reconstructed video quality versus extra bit rate in the pulse for (a) Carphone video sequence at baseline 31 kbps, (b) Container video sequence at baseline 30 kbps

between 15 and 31 kbps. This is supplemented by an occasional high-bandwidth pulse lasting typically 30 milliseconds. In our simulation, these pulses are assumed to occur during the $6^{th}$ and $16^{th}$ frames. The jump update parameter $N$ is set to 10 so the LTR frames are the $1^{st}$ and $11^{th}$ frames, except for the CurrFrameLTR case where the $6^{th}$ and $16^{th}$ frames become the LTR because the algorithm calls for creating a new LTR whenever a pulse of high bandwidth comes along.

**Results:** Sample results are shown in Figure 2(a) which is for the Carphone sequence with a baseline bit rate of 31 kbps, and Figure 2(b) which is for the Container sequence at 30 kbps. Similar results to these were obtained for the other 3 sequences, and for baseline bit rates down to 15 kbps. In these plots, the y-axis shows the average PSNR of the reconstructed video, obtained by computing the mean-squared error of each frame, averaging these values over frames, and then converting to PSNR. The x-axis shows the additional rented bit rate. For any one curve, the quality at one point and at another point represent performance at different bit rates. However, for a given value of the x-axis, one can compare the PSNR values on the four different curves, as these represent the same total bit rate, with four different coding schemes.

We see that CurrFrameLTR outperforms all the other cases, and is approximately 0.5 dB better than the next best approach. The worst performing are oldLTR and LTR+Curr, which involve providing all or some of the extra bits to the LTR frame in the past. Spending bits on improving a past frame does not contribute to the PSNR of that past frame, since the frame is displayed at the decoder before these extra bits arrive. By serving as a high-quality reference, this improved LTR only marginally helps with the performance of future frames. It is possible that for particular sequences, enhancing a past LTR frame would be beneficial if some object that was present in the LTR frame and is absent in the current frame, reoccurs in the next few frames. Also, if the extra bandwidth in the pulse is so high that adding extra bits does not provide significant PSNR improvement to the current frame, then it would be advisable to give those extra bits to the past LTR frame. However, in general, the conclusion of this section is that the best strategy for a dual frame encoder which is presented with occasional pulses of extra bandwidth is to deploy the extra bits on the current frame, thereby making it a high quality frame, and to signal the decoder that this current frame is to be retained as the long term reference.

# 3 Scenario with a Lookahead Buffer

We now consider a scenario where the encoder has a lookahead buffer of $B$ frames, thereby incurring some buffering delay. There are $B$ frames stored in the encoder input buffer, including the current frame to be encoded and $B-1$ future frames. We assume that the short pulse of high rental bandwidth may be used to encode the current frame only, or to encode the current and some of the future frames in the buffer.

In a cognitive radio scenario, a legacy user such as a fire department who owns a particular band of spectrum might rent out the spectrum by the second or by the minute when it is not in use. The legacy user might begin to transmit at any time, in which case a rental agreement that is in progress would be cancelled. The renter would then have to look elsewhere to rent spectrum. This scenario presents the video encoder with a situation where pulses of high bandwidth may occur fairly frequently, but fairly unpredictably, and high bandwidth cannot be relied on to remain available, but can disappear on short notice. In this scenario, during a pulse of high bandwidth, the video encoder may wish to spend some of the available bits on *future* frames in the lookahead window, because there is no guarantee that there will be high bandwidth available in the future. We saw in the previous section that, in the absence of a lookahead window, the encoder should use the high bandwidth on the current frame, and make it the LTR frame. Here, in the presence of a lookahead buffer, we envision four ways to use the high-bandwidth pulse:

- **LTR all**: Give all extra bits to the current frame only, and assign the current frame as the LTR frame. This is the same as the case of CurrFrameLTR of the previous scenario, which was the best approach among the ones with no lookahead buffer.

- **LTR+Equal**: Give sufficient bits to the current frame so that it achieves some specified level of high quality, and distribute the remainder of the bits equally among the other frames in the lookahead buffer. Assign the current frame to be the LTR frame.

- **LTR+Unequal**: Give sufficient bits to the current frame to achieve high quality and distribute the remainder of the bits among the other frames unequally, giving more to the last frame in the window. In this way, the benefits of the high bandwidth pulse can be stretched into the future because the high quality of that last frame can propagate into subsequent frames by the mechanism of short-term prediction. Assign the current frame as the LTR frame.

- **LTR+Unequal+Reset**: This is same as LTR+Unequal but where the last frame of the buffer is set to be the LTR frame for the frames that come after. (So, although the jump update parameter would dictate that the LTR frame is updated every N frames, here we update the LTR frame to be the $1^{st}$ frame in the buffer, and again we update it to be the last frame in the buffer, and then we return to the schedule of updating every $N^{th}$ frame). Here, the benefits of the high bandwidth pulse are stretched even farther into the future because the high quality of the last frame can be used by subsequent frames by the mechanism of long-term prediction.

The first strategy here is to use all the bits for the current frame. For all the other strategies, some of the extra bits are used to ensure that the current frame has some sufficient level of high quality, and the remaining bits are used to encode other frames in the lookahead buffer. The reasoning is that, while having a high-quality long-term reference frame provides an advantage for the entire set of frames that reference it, providing ultra-high quality to the LTR frame will produce diminishing returns. We have conducted some simulations to determine what this high quality level should be.

The effect of the LTR frame quality on future frames is shown in Figure 3. In this experiment, 30 frames were encoded, and the LTR frame (frame 20) is encoded with varying quality by transmission of extra bits in its enhancement layer. Its effect was observed for the next 9 frames. In this plot, the x-axis shows the PSNR of the LTR frame and the y-axis represents the average PSNR of the subsequent 9 frames. Two of the sequences (the low-motion sequences Claire and Mother-daughter) show diminishing benefits for the frames following the LTR frame when the LTR frame has a quality higher than about 50 dB. For the other three sequences (Foreman, Container, and Carphone) there is little gain to be had by pushing the LTR quality above 45 dB. So for the three cases LTR+Equal, LTR+Unequal, and LTR+Unequal+Reset, we allocate enough bits to the current frame so that its PSNR reaches approximately 45 dB, and the remaining bits are allocated to the other frames in the buffer.

For the LTR+Equal case, the remaining bits are allocated equally among the other frames in the buffer. We note that, following a high-quality LTR frame, the subsequent frames can be compressed with high quality using relatively few bits. For the LTR+Unequal case, after the current frame is encoded to reach the threshold, we allocate few bits to the frames in between the first and last buffered frames. The remaining bits are given to the last frame. Specifically, about 40% of the remaining bits are given to the last frame for the simulation with a large pulse of extra bits, and about 60% of the remaining bits are given to the last frame for the smaller pulse of extra bits, which in both cases causes the last frame in the buffer to be about 2 dB higher than the last frame in the LTR+Equal case. This distribution has been obtained experimentally. In this way, we expect relatively high quality to be maintained for all frames currently in the buffer and in addition, the high quality of the last frame

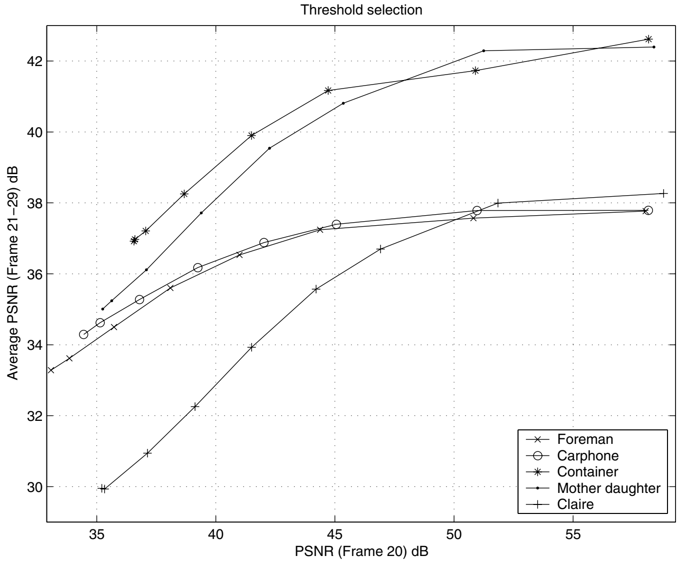


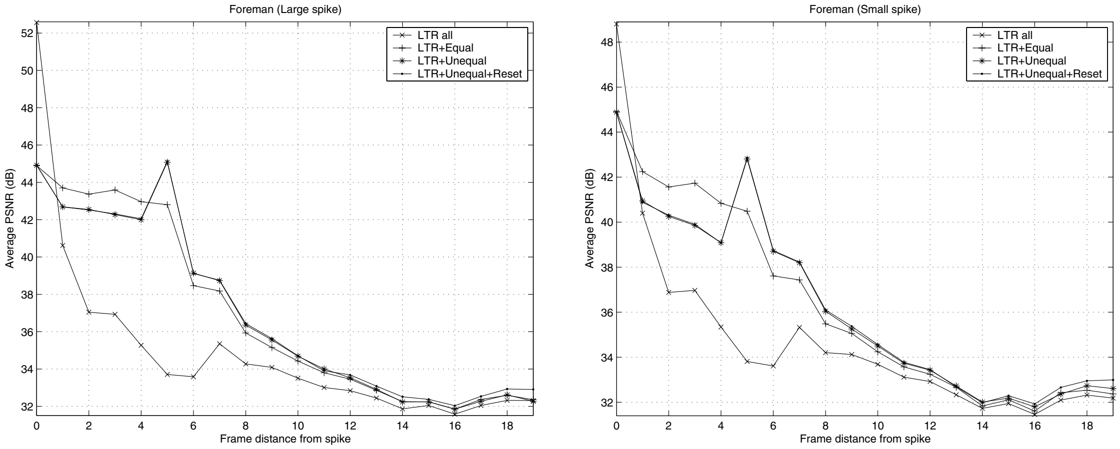
Figure 3: Threshold selection process: the PSNR of the 9 frames that follow the LTR frame is plotted versus the PSNR of the LTR frame, and we see diminishing returns.

will last for some more time, which may be helpful in case the extra bandwidth drops unexpectedly by return of a legacy user.

**Results:** The experimental setup is similar to that described previously. The frame rate was 30 fps and the initial jump update parameter, $N$, was 20. A total of 80 frames are encoded and high bandwidth pulses are considered to be at frame numbers 21, 41, and 61. The frame buffer is of size 6.

Figure 4 shows the results for the Foreman video sequence. The x-axis represents the distance of a frame from the frame where a pulse occurred. The y-axis represents the average PSNR for that frame after reconstruction. Figure 4(a) is for a large pulse of extra bits (98.8 kilobits) and Figure 4(b) is for a smaller pulse (68.4 kilobits). In the LTR all case, the PSNR is very large where the pulse occurs since all the bits are used up there. The benefit of this pulse dies down rapidly. The LTR+Equal approach pegs the current frame quality at 45 dB and distributes the remaining bits over the next 5 frames. The quality of these frames is higher than for the LTR all case. In the LTR+Unequal case, after starting at the same 45 dB level as in LTR+Equal, the quality of the first few frames in the buffer is kept slightly lower than the LTR+Equal in order to save bits to produce high quality for the last frame in the buffer. By doing so, higher quality is maintained for a few additional frames. A very tiny additional gain comes from resetting the LTR frame and assigning the last buffered frame as the new LTR frame for the next $N$ frames or until a new spike occurs. Similar results were found for spikes of various sizes.

The effect of the pulse size on the average PSNR for the whole sequence is shown in Figure 5. The x-axis shows the average pulse size and the y-axis represents the PSNR averaged over the whole sequence. LTR all is almost flat because all bits are



(a) Foreman with large pulse

(b) Foreman with small pulse

Figure 4: PSNR of reconstructed video quality versus frame distance from the high-bandwidth pulse for (a) Foreman video sequence with average pulse size of 98.8 kbits, (b) Foreman video sequence with average pulse size of 68.4 kbits

used for the current frame, and quality over the 45 dB threshold produces negligible improvement in subsequent frames. For the other cases, the PSNR improves as the pulse size increases.

The same experiment was also performed with random locations of spikes. If there was no bandwidth spike in the last 20 frames, the encoder enters a bandwidth-seeking mode in which it attempts to find extra bandwidth every 5 frames. With probability 0.7, the attempt is successful, in which case the encoder does not search for extra bandwidth for another 20 frames. If the attempt fails, the encoder continues trying once in the next 5 frames, and so on. We performed this experiment for 150 frames of the Foreman sequence. The trends of results are the same as shown in Figure 4.

# 4 Conclusions and Future Work

We are concerned with the scenario of video transmission in a cognitive radio environment. We suppose that the video transmitter has a constant low-rate baseline bandwidth, but that the video transmitter can supplement this on occasion by renting extra spectrum for short or long periods of time. Even when rented for a long period of time, the rental agreement risks being cancelled. We envision a situation where the transmitter will choose to scavenge occasional pulses of bandwidth which may be low cost because they are short, and because they are at imminent risk of being cancelled when the legacy user decides to use that piece of spectrum.

Under these conditions, we first considered a video encoder with no lookahead buffer, and we showed that a dual frame video encoder could profitably make use of

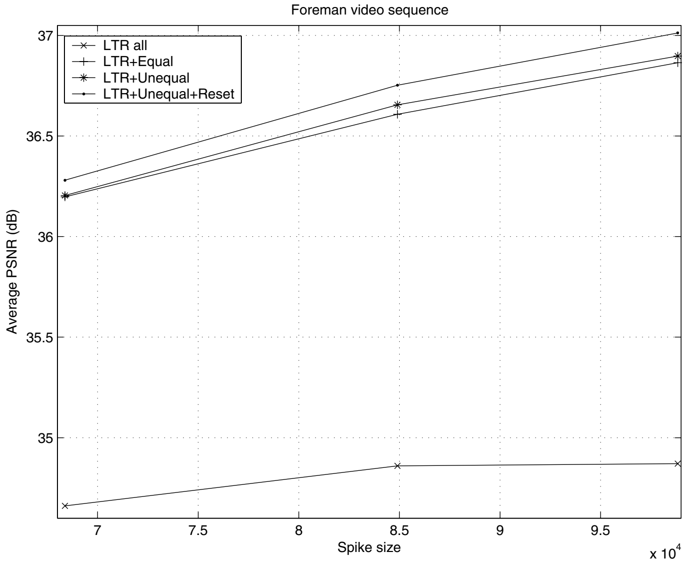


Figure 5: Variation of average PSNR with spike size for the Foreman video sequence

even a very short pulse of bandwidth by creating a high quality frame and using this frame as a long-term reference frame for some period of time. We did not find any advantage to deploying bits to enhancing frames in the past.

We next considered a video encoder with a lookahead buffer, and showed that the encoder could get further advantage by using only some of the bits for the current frame, and distributing some of the bits to the other frames in the buffer. In particular, providing bits to the current frame produces diminishing returns in terms of its value as a long term reference frame.

In the future, we would like to study how these various approaches compare as one varies the size of the delay buffer. Also we would like to optimize the allocation of bits to frames for varying sizes and durations of the pulses.

**Acknowledgment:** This research was supported in part by the UC Discovery Grant Program of the State of California, and by the Center for Wireless Communications.

# 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, Feb. 1997.

[2] G. Côté, B. Erol, M. Gallant, and F. Kossentini, "H.263+: Video Coding at Low Bit Rates," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 7, pp. 849–865, Nov. 1998.

[3] 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*, July 2002.

[4] M. Gothe and J. Vaisey, "Improving Motion Compensation Using Multiple Temporal Frames," *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing,* vol. 1, pp. 157–160, 1993.

[5] M. Budagavi and J.D. Gibson,"Multiframe Video Coding for Improved Performance Over Wireless Channels," *IEEE Transactions on Image Processing,* vol. 10, pp. 252–265, Feb. 2001.

[6] T. Wiegand, X. Zhang, and B. Girod, "Long-Term Memory Motion-Compensated Prediction," *IEEE Transactions on Circuits and Systems for Video Tech.*, vol. 9, pp. 70–84, Feb. 1999.

[7] T. Fukuhara, K. Asai, T. Murakamai,"Very Low Bit-Rate Video Coding with Block Partitioning and Adaptive Selection of Two Time-Differential Frame Memories," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 3, pp. 212–220, Feb. 1997.

[8] A. Leontaris and P. Cosman, "Video compression with intra/inter mode switching and a dual frame buffer," *IEEE Data Compression Conference 2003*, pp. 63-72, 2003.

[9] A. Leontaris and P. Cosman, "Dual Frame Video Encoding with Feedback," *Proceedings 37th Asilomar Conference on Signals, Systems, and Computers*, pp. 1514-1518, Nov 2003.

[10] A. Leontaris and P. Cosman, "Video Compression for Lossy Packet Networks with Mode Switching and a Dual-Frame Buffer," *IEEE Transactions on Image Processing,* Vol. 13, No. 7, pp. 885-897, July 2004.

[11] V. Chellappa, P.C. Cosman, and G.M. Voelker, "Error Concealment for Dual Frame Video Coding with Uneven Quality Assignment," *IEEE Data Compression Conference 2005*, pp. 319-328, Mar 2005.

[12] V. Chellappa, P.C. Cosman, and G.M. Voelker, "Dual Frame Motion Compensation With Uneven Quality Assignment," *IEEE Data Compression Conference 2004*, pp. 262-271, Mar 2004.

[13] V. Chellappa, P.C. Cosman, and G.M. Voelker, "Dual Frame Motion Compensation for a Rate Switching Network," *Proceedings 37th Asilomar Conference on Signals, Systems, and Computers*, pp. 1539-1543, Nov 2003

[14] J. Mitola III and G.Q. Maguire, Jr. "Cognitive Radio: Making Software Radios More Personal," *IEEE Personal Communications*, vol. 6, issue 4, pp. 13-18, Aug. 1999.

[15] J. Mitola III, "Cognitive Radio for Flexible Mobile Multimedia Communications," *Mobile Networks and Applications, Kluwer Academic Publishers*, vol. 6, no. 5, pp. 435-441, Sept 2001.

[16] "SDR Forum Website," *http://www.sdrforum.org*, 1999.

[17] J. Walko, "Cognitive Radio," *IEE Review*, vol. 51, issue 5, pp 34-37, May 2005.

[18] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560-576, July 2003.

[19] "H.264/AVC Reference Software," *http://iphome.hhi.de/suehring/tml/*.