

UC San Diego

UC San Diego Previously Published Works

Title

Delay Constrained Multiplexing of Video Streams Using Dual-Frame Video Coding

Permalink

<https://escholarship.org/uc/item/612894cc>

Journal

IEEE Transactions on Image Processing, 19(4)

ISSN

1057-7149 1941-0042

Authors

Tiwari, M.
Groves, T.
Cosman, P. C.

Publication Date

2010-04-01

DOI

10.1109/TIP.2009.2038812

Peer reviewed

Delay Constrained Multiplexing of Video Streams Using Dual-Frame Video Coding

Mayank Tiwari, *Student Member, IEEE*, Theodore Groves, and Pamela C. Cosman, *Fellow, IEEE*

Abstract—We consider the multiplexing problem of transmitting multiple video source streams from a server over a shared channel. We use dual-frame video coding with high-quality Long-Term Reference (LTR) frames and propose multiplexing methods to reduce the sum of mean squared error for all the video streams. This paper makes several improvements to dual-frame video coding. A simple motion activity detection algorithm is used to choose the location of LTR frames as well as the number of bits given to such frames. An adaptive buffer-constrained rate-control algorithm is devised to accommodate the extra bits of the high-quality LTR frames. Multiplexing of video streams is studied under the constraint of a video encoder delay buffer. Using H.264/AVC, the results show considerable improvement over baseline schemes such as H.264 rate control when the video streams are encoded individually and over multiplexing methods proposed previously in the literature. The high-quality LTR frames are offset in time among different video streams. This provides the benefits of dual-frame coding with high-quality LTR frames while still fitting under the constraint of an output delay buffer.

Index Terms—Dual-frame buffer, H.264/AVC, high-quality updating, long-term reference frame, video compression.

I. INTRODUCTION

TRANSMISSION of multiple video streams from a central server (or from multiple servers but with centralized rate allocation) to multiple destinations over a path with a shared link is a familiar scenario in many applications. Some applications are Direct Broadcast Satellite (DBS), transmission of digital video over wireless broadband, video-on-demand services, video surveillance, and transmission in a cognitive radio situation where multiple users share the same bandwidth. In DBS, many video streams are compressed and transmitted together from a satellite to different receivers, and all the video streams share the same bandwidth. As digital video compression technology becomes more efficient, more video streams can be compressed and transmitted together. The total bit-rate of

multiple video streams is limited by the bandwidth of the central server. Equally distributing the available resources among the video streams often produces a poor result. Therefore, it is important to efficiently allocate the overall bit-rate among the compressed video streams at every time instant to enhance the overall quality.

Fig. 1 shows independent rate control for multiple video streams. Each encoder generates a variable bit-rate stream. Each encoder maintains a separate output buffer to convert its output stream to a constant bit-rate stream. Based on the output buffer fullness and the complexity of the frames, each encoder maintains a separate rate control path to encode its video. All the bitstreams are multiplexed together and transmitted through a constant bit-rate channel. At the decoder, the bitstreams are demultiplexed and each bitstream is sent to the input buffer for its decoder. Each decoder sequentially fetches its bitstream from its input buffer, decodes the frame, and sends it to an output display buffer. Fig. 2 shows a method of joint bit-rate allocation for multiple video streams. Here, an output buffer is shared by all the video encoders. All the videos are encoded separately and each encoder passes some information to a central controller. Based on the information from each encoder about its video complexity and the status of the combined output buffer, the central controller decides the number of bits that should be allocated to each video stream. The video encoders use this information to encode their video which is then stored in the output buffer. The output buffer then sends the encoded bitstream through a constant bit-rate channel to the decoder. At the decoder, the bitstream is buffered and demultiplexed. Each bitstream is decoded separately and sent to its output display buffer.

Rate control algorithms for encoding video streams independently were extensively studied [1], but joint bit-rate allocation has been widely used to improve the overall quality for multiple video streams [2]–[9]. A multicamera surveillance system was considered in [2] where Peak Signal-to-Noise Ratio (PSNR) improvement was shown for transmitting video content only when there is any activity captured by a camera. However, it did not consider the case in which all cameras were capturing activity simultaneously. A distributed approach with high-convergence time for transmitting multiple video streams was considered in [3] where the bit-rate allocation was done by the link price which is updated using the subgradient method. A parallel encoder system with large delay and memory requirements was adopted in [4] where multiple streams are encoded with several bit-rates, and a combination of bit-rates for multiple streams was selected to maximize the PSNR. A superframe concept was used in [5] where one frame each from multiple video streams is combined into a superframe, and a Quantization Parameter (QP) is found based on the relative complexity of the superframes to

Manuscript received January 06, 2009; revised September 15, 2009. First published December 18, 2009; current version published March 17, 2010. This work was supported in part by the Office of Naval Research, in part by the National Science Foundation under Grant 0635165, in part by the Center for Wireless Communications at UC San Diego, and in part by the UC Discovery Grant program. Part of this work was presented at the IEEE International Conference on Acoustic, Speech, and Signal Processing, March 2008. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sohail Dianat

M. Tiwari and P. C. Cosman are with the Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92093-0407 USA (e-mail: mayank@ece.ucsd.edu; pcosman@ucsd.edu).

T. Groves is with the Department of Economics, University of California, San Diego, La Jolla, CA 92093-0508 USA (e-mail: tgroves@ucsd.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIP.2009.2038812

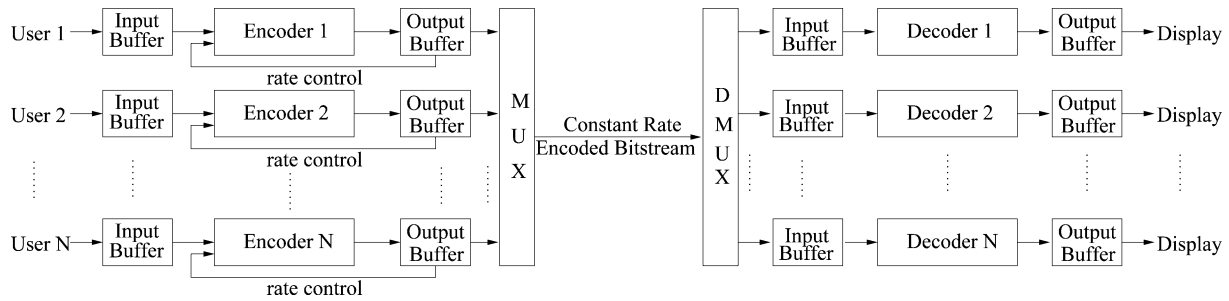


Fig. 1. Multiple video streams with separate output buffers and rate control paths for each user.

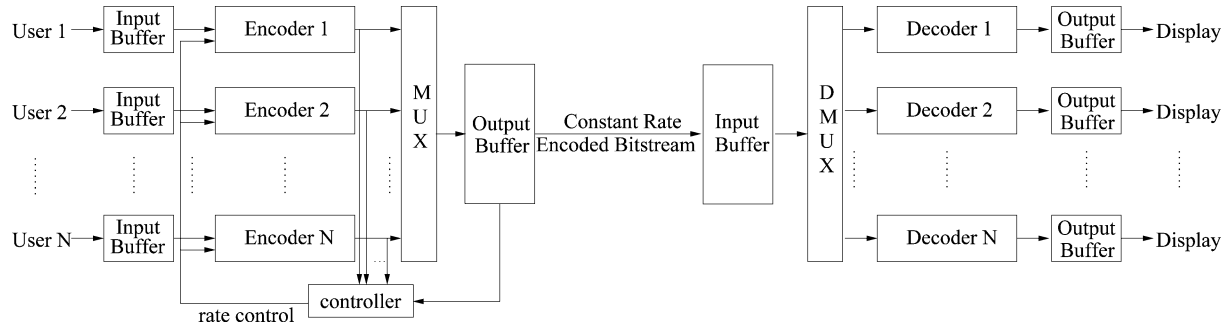


Fig. 2. Bit-rate allocation for multiple video streams with a common output buffer and rate control path for all the users.

improve the overall PSNR. In [6], a better joint rate control algorithm was proposed for the superframe method. In [7], a resource allocation algorithm was proposed to reduce PSNR fluctuation while maintaining high PSNR using Fine Granularity Scalability (FGS). It reduces PSNR fluctuations but also reduces overall PSNR. A closed form solution in the transform domain was proposed in [8] that minimizes the distortion variance with a small coding penalty. A joint rate control algorithm to dynamically distribute the channel bandwidth among multiple video encoders was proposed in [9] with the objective of assigning approximately equal quality to all videos.

In [10], three optimization objectives for transmitting multiple video streams over a shared channel were studied: maximizing overall PSNR, minimizing overall Mean Squared Error (MSE), and minimizing the maximum MSE. Using subjective tests, it was shown in [10] that minimizing the overall MSE corresponds best to subjective preferences. Using this result from [10], we chose the performance criterion of minimizing the overall MSE. The results here should not be compared directly with a method that assumes any other performance criterion.

In this paper, we propose joint bit-rate allocation methods for multiple video streams. The video Rate-Distortion (R-D) information from each encoder is sent to a controller. Based on the R-D information for the videos and the status of the output buffer, the controller calculates the optimal operating point for each video and sends the number of bits allocated to each encoder. In addition to the multiplexing methods, we use dual-frame video coding [11]–[14] to further improve the video quality by carefully selecting the location and number of bits given to the LTR frames. We use the motion of the video stream to find the location and the number of bits assigned to the LTR frames [15]. Rate control generates variable size bitstreams for each frame in a video. Dual-frame video coding

with high-quality LTR frames further increases the variation in bitstream size. Therefore, an encoder output buffer is required to convert such a variable stream to a constant stream so that it can be transmitted through a constant rate channel without causing frame losses. The encoder output buffer increases the delay in overall transmission, which can be a problem for real-time video transmission. Buffer constrained rate control for a video stream using dual-frame video coding was studied in [16]. We extend this concept to multiple video streams where an encoder output buffer is shared among various users as in Fig. 2. We compare our multiplexing methods against rate allocation using (a) H.264 JM rate control, (b) dual-frame video coding, and (c) the superframe methods described in [5], [6]. While there are many coding variations that can yield improved compression performance, the use of high-quality LTR frames in dual-frame video coding not only improves performance for individual videos, but also has a particular advantage in a buffer constrained multiplexing situation because the high-quality LTR frames which consume a large share of the delay buffer can be staggered among the different multiplexed streams.

For dual-frame video coding, the main contributions of our paper over previous methods are as follows: (a) we use motion activity to assign the number of bits to LTR frames which performs better than the fixed number of bits allocated to LTR frames in previous work, (b) the LTR frame locations in dual-frame video coding are chosen adaptively using the activity measurement and this outperforms dual-frame video coding with evenly spaced LTR frames, and (c) we design a rate control algorithm for dual-frame video coding with high-quality LTR frames and our algorithm outperforms the conventional rate control algorithms as well as rate control algorithms previously designed for dual-frame video coding.

For video multiplexing, the main contributions of our paper over the previous methods are as follows. (a) None of the

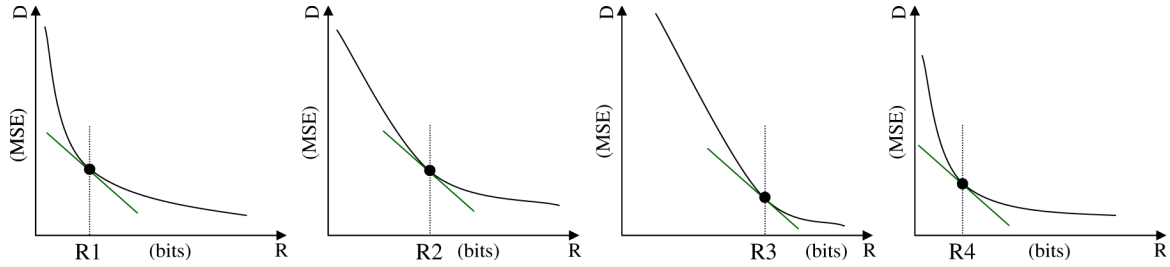


Fig. 3. STR_ES method for four videos. R1, R2, R3, and R4 are the number of bits allocated to each video. At this allocation, the slopes of the R-D curves are equal.

previous methods in video multiplexing used dual-frame video coding. The previous methods of video multiplexing allocated more bits to a video with high motion by taking bits from low motion videos. Therefore, the low motion video quality suffers. Our dual-frame video coding technique gives huge MSE reduction for low motion videos compared to high-motion videos. Therefore, we combine our dual-frame video coding method with the existing methods of allocating bits based on the motion. In our method, the LTR frames receives the bits based on the activity measurement while the remaining frames are allocated bits based on the relative motion between the videos. The new multiplexing method further reduces the overall MSE. (b) The buffer-constrained rate control for dual-frame video coding was modified to accommodate the high-quality LTR frames from all the video streams in order to avoid buffer overflow.

This paper is organized as follows. Section II discusses multiplexing methods where the bits are allocated to multiple video streams depending on the relative complexity of the videos. Section III describes several multiplexing methods using dual-frame video coding with high-quality LTR frames. Section IV introduces the delay constrained rate control for dual-frame video coding and the modifications in rate control to accommodate the high-quality LTR frames. This section suggests a rate control modification to all the multiplexing methods discussed in the previous sections. The results are discussed in Section V. Section VI concludes the paper and provides future directions.

II. MULTIPLEXING VIDEO STREAMS WITH NO ENCODER OUTPUT DELAY

We start with simple methods for multiplexing video streams using dual-frame video coding in which we assume a negligible size of encoder output buffer. In multiple frame prediction, more than one past frame can be 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 [17], [18]. Recent video standards such as H.264/AVC allow the use of up to 16 reference frames for encoding. However, video quality improvement diminishes with the increase in the number of reference frames. We encode using two reference frames, called dual-frame video coding. The reference frames can be any encoded frames prior to the current frame. We use a simple form of dual-frame video coding for the multiplexing methods described in this section where two immediate past frames are used

as reference frames to encode the current frame. We call these frames Short-Term Reference (STR) frames.

Suppose we have N video streams and B kbps of total bit-rate available to transmit these video streams. The simplest bit-rate allocation is to divide bits equally among the video streams and among the frames. If a video stream is f frames per second, then each frame gets $B/(N \cdot f)$ bits. Each video stream is encoded with dual-frame video coding with two STR frames. We call this method **STR_EB** and it could be called a “fair” allocation. Given the number of bits to encode a frame of a video stream, the reference software model of H.264 [19] will search and choose the best prediction mode to reduce the MSE.

Better multiplexing exploits the relative complexity (R-D properties) of video streams. We take the curve-fitting model for the R-D curve of a frame in video stream n to be

$$D_n(R_n) = a_n + \frac{b_n}{R_n} \quad (1)$$

where R_n is the number of bits, D_n is the distortion for a frame in video stream n , and a_n, b_n are the curve-fitting coefficients found using least squares. Other curve-fitting models are available in the literature [20]. We generate R-D measurement points using 14 different QPs (ranging from 10 to 51) and that was found to be sufficient to calculate the curve-fitting coefficients for a broad range of bit-rates (ranging from 3 kbps to 1500 kbps for QCIF videos, depending on the video complexity). The complexity of generating R-D curves can be reduced by using the method described in [21] and is not included in this paper.

Given the R-D curve-fit for a frame in each video stream, the sum of MSE for all the video streams can be minimized using standard optimization techniques such as the Lagrangian multiplier [22]. Consider the i^{th} frame of each video stream. The optimization problem can be formulated as

$$\min_{R_n} \sum_{n=1}^N D_n(R_n) \quad \text{subject to} \quad \sum_{n=1}^N R_n \leq \frac{B}{f}. \quad (2)$$

Using Lagrange multipliers, the bit allocation for video j is

$$R_j = \frac{\sqrt{b_j}}{\sum_{n=1}^N \sqrt{b_n}} \times \frac{B}{f}, \quad \forall j \in 1, 2, \dots, N. \quad (3)$$

The bit allocation achieved by (3) essentially finds a point in each R-D curve where the slope is the same for all the curves.

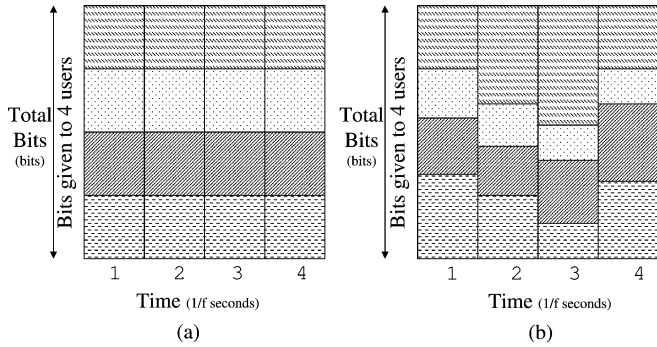


Fig. 4. Multiplexing methods for four video streams: (a) STR_EB; (b) STR_ES.

Known as the *equal slope* technique, this is shown in Fig. 3. After bit allocation, the videos are encoded with dual-frame coding with two STR frames. We denote this method of bit-rate allocation **STR_ES**. Although not identical, the basic approach is described in [10]. This minimizes, on a per frame basis, the sum of MSEs for all the video streams. This method gives extra bits to a video stream that is experiencing high motion by taking some bits from the low motion streams. STR_ES will result in STR_EB allocation if the R-D curves for all the video streams are the same, in which case this method will allocate $B/(N \cdot f)$ bits for each stream. The R-D curve exists only at certain discrete points (because the QP takes on discrete values). Therefore, it may not be possible to achieve the exact bit-rate for a video specified by (3). The STR_ES method chooses the R-D point that is closest to the bit allocation determined by (3). Ideally, we do not need an output buffer for these two multiplexing methods to store the encoded bitstream to be transmitted (because the multiplexed bitstreams achieve the constant target output rate for each frame). In practice, however, because of the discrete nature of the R-D curve, we need a small output buffer to accommodate the difference between the target rate for a frame and the actual bit-rate achieved at some QP.

These two methods of multiplexing are shown in Fig. 4. Each shaded pattern in the figure represents one video stream, and each block represents the size in bits of a frame. On the y axis is the number of bits given to each video and on the x axis is the time slot, assuming a time slot of $1/f$ seconds at f frames per second. In Fig. 4(a), each video in each time slot gets the same number of bits, so the bits per user per time slot can be depicted as identical boxes. Fig. 4(b) shows the STR_ES method where the videos with higher activity take bits from the ones with lower activity. The videos do not take bits from any other time slot, so the depiction of total bits per time slot still shows vertical boundaries. Fig. 5 shows the flow chart for the (a) STR_EB method and (b) STR_ES method. We expect STR_ES will perform better than STR_EB because, in STR_ES, bits are allocated to different video streams based on their complexity.

We compared our methods for multiplexing video streams with the superframe method given in [5]. We applied the superframe method with the total number of bits at each frame as discussed in STR_EB. This method is denoted by **SF_EB**. For a fair comparison, we use two STR frames for the SF_EB method so it has the same advantage of dual-frame video coding. The

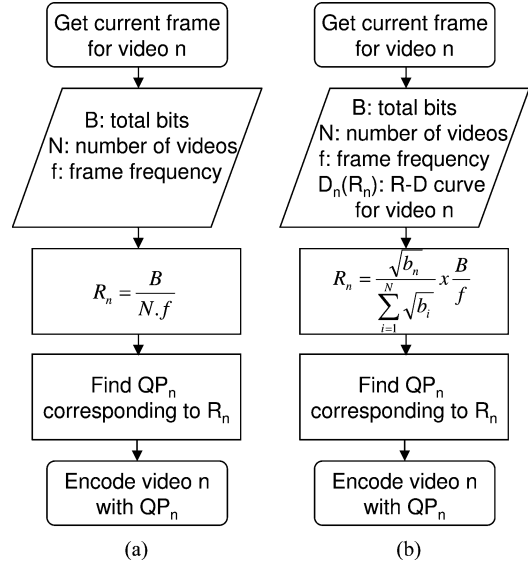


Fig. 5. Flow chart for the (a) STR_EB and (b) STR_ES multiplexing methods.

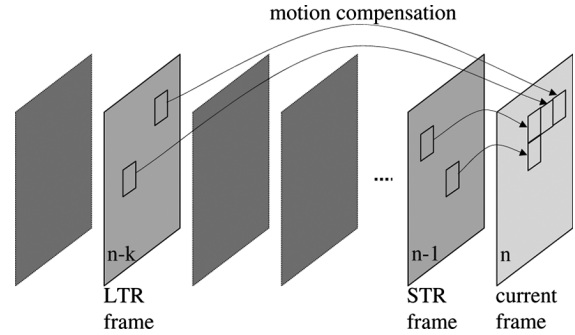


Fig. 6. Dual-frame video coding with one short-term and one long-term reference frame.

picture in Fig. 4(b) applies to this method, except that the block boundaries are now defined by assigning the same QP for all the videos instead of the same slope.

III. MULTIPLEXING VIDEO STREAMS USING ADAPTIVE DUAL-FRAME VIDEO CODING

In the previous section, we used dual-frame coding where two frames immediately preceding the current frame to be encoded were used as references. It was shown in [11] and [12] that temporally separated reference frames perform better than consecutive reference frames.

Since adjacent frames are temporally correlated, it is useful to have the immediate past frame as a reference. In addition, some frame already encoded in the past can be chosen as a reference. These two frames [13], [14], are called the STR and Long-Term Reference (LTR), as shown in Fig. 6. Both encoder and decoder store LTR and STR frames. For encoding frame n , the STR is frame $n-1$ and the LTR is frame $n-k$, for some $k > 1$. The LTR frame can be chosen by jump updating, in which the LTR frame remains the same for encoding K frames, then jumps forward by K frames and again remains the same for encoding the next K frames. In jump updating, every frame serves as an STR,

but only every K^{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 was shown to enhance the quality of the entire stream [13], [14]. In this section, we will use dual-frame video coding with high-quality LTR frames for enhancing the quality of multiple video streams simultaneously. We use the motion of a video sequence to determine the number of bits given to an LTR frame.

A. Bit Allocation for LTR Frames

In dual-frame video coding, one key issue is to allocate an appropriate number of bits to ensure a high-quality LTR frame. For low motion videos, we should allocate many bits to the LTR frame since subsequent frames are similar to the LTR frame and will benefit from the high quality of the LTR frame. For high-motion parts, it is not desirable to spend many bits on an LTR frame because its higher quality will soon be useless as the subsequent frames rapidly become different from the LTR. We use the activity in a video to determine the quality given to an LTR frame.

To measure activity, we divide a frame into MacroBlocks (MBs) of standard size (16×16 pixels) and calculate the pixel-by-pixel Sum of Absolute Differences (SAD) between each MB and the co-located MB in the previous frame. For MB k in the current frame n ($MB_{n,k}$)

$$SAD(n, k) = \sum_{i=1}^{16} \sum_{j=1}^{16} |MB_{n,k}(i, j) - MB_{n-1,k}(i, j)|. \quad (4)$$

The MB is considered to be active if $SAD(n, k) > T$, for some predetermined threshold T . We chose $T = 500$ after examining a range of thresholds for various QCIF video sequences. Activity measurement is done in real-time. A similar method in [2] considered binary classification of activity for video surveillance. Note that motion vectors can also be used to perform the motion activity detection.

A larger number of extra bits (beyond those normally assigned to non-LTR frames) are assigned to an LTR frame for a low motion part of a video because the high quality of LTR frames will be retained for a long time. On the other hand, fewer extra bits are assigned to a high-motion part of a video because the high quality will soon be lost and a new LTR frame will soon be needed. To operate in real-time and avoid buffering future frames, we consider the motion of past frames to predict the motion of future frames. Let m be the average fraction of active MBs in the 10 frames prior to an LTR frame. Based on m , the bit allocation for the LTR frame (LTR_bits) is given by

$$LTR_bits = \begin{cases} 2 \times reg_bits, & \text{if } m > 0.5 \\ 10 \times reg_bits, & \text{if } m < 0.1 \\ (12 - 20 \times m) \times reg_bits, & \text{otherwise} \end{cases} \quad (5)$$

where reg_bits is the average number of bits assigned to a regular frame. These allocations and threshold values were determined experimentally and not carefully optimized for any particular video stream. Improvement was found for nearly all of the video streams [15] compared to having a fixed allocation of high quality for LTR frames.

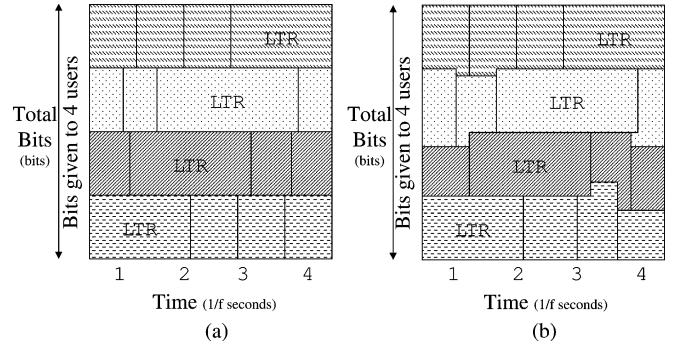


Fig. 7. Multiplexing methods for four video streams (a) eLTR_EB and (b) eLTR_ES.

B. Multiplexing Video Streams Using Dual-Frame Video Coding With High-Quality LTR Frames

Both the multiplexing methods in the previous section use dual-frame video coding with two STR frames for motion compensation. We can further reduce the sum of MSE by exploiting the high-quality LTR frame in dual-frame video coding. Let L_n be the number of bits assigned to an LTR frame for the n^{th} video stream. Note that L_n varies with the motion of a video stream, as described previously. The extra bits given to the LTR frame are taken equally from the regular frames between two high-quality LTR frames. Let K be the distance between two LTR frames. Using STR_EB, each video stream should get $(K \times B)/(N \times f)$ bits for K frames. In the high-quality LTR extension to STR_EB, out of this pool of bits, L_n bits are assigned to an LTR frame. The remaining bits $((K \times B)/(N \times f) - L_n)$ are equally divided among each of the remaining $K - 1$ frames in that group of K frames in the stream. If r_n denotes the number of bits allocated to each of these $K - 1$ frames, then

$$r_n = \frac{\frac{K \times B}{N \times f} - L_n}{K - 1}. \quad (6)$$

This may also be deemed a “fair” allocation since each video stream receives an equal number of bits for the entire video. Although the number of bits L_n given to the LTR for stream n may be higher than that for some other stream, the number of bits r_n given to the other $K - 1$ frames in that group of K frames for stream n will be correspondingly lower, so each video stream is allocated an equal number of bits over all frames. This method is called **eLTR_EB**.

We can also incorporate dual-frame video coding with high-quality LTR frames in the STR_ES method. Again, the bit allocation for LTR frames is as described in eLTR_EB. Using (3), for any frame where no stream has an LTR, the bits allocated to video j are

$$R_j = \frac{\sqrt{b_j}}{\sum_{n=1}^N \sqrt{b_n}} \times \sum_{n=1}^N r_n, \quad \forall j \in 1, 2, \dots, N \quad (7)$$

where r_n is defined by (6). If, at any time instant, a video stream k has an LTR frame, then that video stream receives L_k bits and

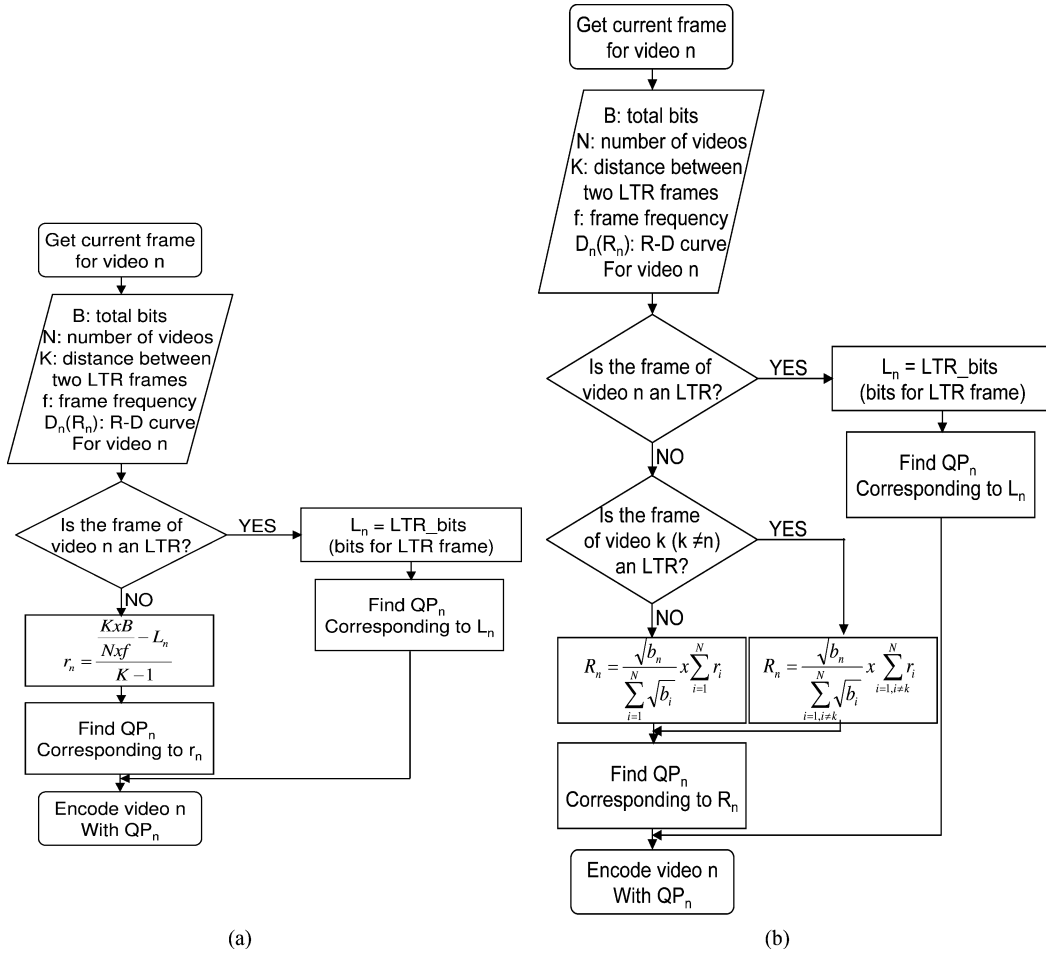


Fig. 8. Flow chart for the (a) eLTR_EB and (b) eLTR_ES multiplexing methods.

the remaining video streams will receive an allocation similar to STR_ES, i.e.,

$$R_j = \frac{\sqrt{b_j}}{\sum_{n=1, n \neq k}^N \sqrt{b_n}} \sum_{n=1, n \neq k}^N r_n, \quad \forall j \in 1, 2, \dots, N, j \neq k. \quad (8)$$

This method uses dual-frame video coding with high-quality LTR frames, and equal slope allocation for regular frames, and is denoted by **eLTR_ES**. Note that both eLTR_EB and eLTR_ES require an output buffer, due to high-quality LTR frames, to store the encoded bitstream for transmission over a constant bit-rate channel. Since the LTR frames are assigned more bits than the regular frames, the chances of buffer overflow are higher for an LTR frame. Irrespective of the number of bits for an LTR frame determined by the motion activity, the size of an LTR frame is always upper bounded by the available space in the output buffer (*avail_buf*), i.e.,

$$\text{LTR.bits} = \min(\text{LTR.bits}, \text{avail_buf}). \quad (9)$$

Fig. 7(a) depicts the eLTR_EB method. Extra bits for the LTR frames of each video are taken from regular (non-LTR) frames of the same video. Since the extra bits for the LTR frames of one video are not taken from any other video, the depiction of bits per user still has horizontal boundaries. In Fig. 7(b),

bits are taken across users and across time slots to accommodate both higher activity videos and LTR frames. Fig. 8 shows the flow chart for the (a) eLTR_EB and (b) eLTR_ES multiplexing method. We expect that eLTR_EB will perform better than STR_EB, and eLTR_ES will perform better than STR_ES, because of the advantages of dual-frame video coding with high-quality LTR frames.

C. High-Quality LTR Frame Selection

In the previous section and in our previous work for multiplexing video streams [15], we considered evenly spaced LTR frames, irrespective of the video content. As illustrated in Fig. 9, it is possible that some of the evenly spaced LTR frames may not be good [23]. Fig. 9 was generated by repeatedly encoding a QCIF size Mother-Daughter video stream with one high-quality LTR frame each time. The x axis shows the frame number that is being chosen as the high-quality LTR frame, and the y axis represents the percentage of MBs of the following k frames ($k = 20, 50, \text{ and } 100$) which choose to reference the LTR frame over the STR frame. For example, to generate the point on the top curve for $x = 40$ (frame number), we encoded the sequence with only frame 40 as a high-quality LTR frame. We then counted how many MBs out of the next 20 frames (frame 42 to 61) referenced the LTR rather than the STR frame. As 9.2% of MBs referenced the LTR, this gives rise to the plotted

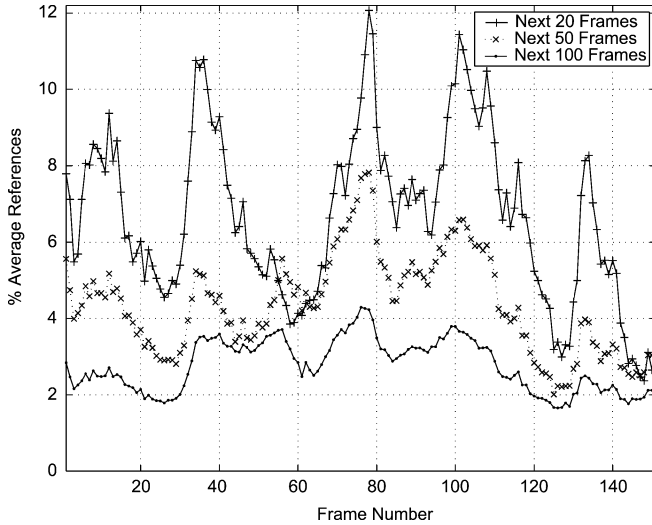


Fig. 9. Percentage average references to a frame when it is created as a high-quality LTR frame in Mother-Daughter video. “Next 20 Frames” shows the effect of an LTR frame on following 20 frames using it.

point (40, 9.2) on the curve for next 20 frames. We found that all the frames are not equally useful as an LTR frame. The frames where we see the peaks (e.g., frames 34, 35, 36, 78) are more useful as LTRs than the frames in the valleys (e.g., frames 24, 25, 26, 60, 61). For example, consider the top curve in the figure which shows, for each possible LTR frame, the percentage of MBs in the next 20 frames which reference the LTR. The plot shows that when frame 78 is chosen to be an LTR, over 12% of the MBs of the next 20 frames prefer to reference it rather than the STR. Therefore, if frame 78 is chosen as an LTR frame then the video quality will likely be high. In contrast, when frame 127 is the LTR, only 3% of MBs in the next 20 frames choose to reference it, which means 97% of the MBs find a better match in the STR. Therefore, if frame 127 is used as an LTR frame then it will not improve the video quality much. So, if we take LTR frames at regular intervals and give them high quality, then they would be ineffective if they fell in such valleys. A method was proposed in [23] using simulated annealing to select an LTR frame that falls at a peak and is effective for subsequent frames. We do not use that procedure in this paper due to its high complexity. A method for LTR frame selection was studied in [24] using color layout descriptors which assumes a large frame buffer at the input to the encoder and the decoder to preselect the possible LTR frames. We do not use that method to select LTR frames because it requires either a standard incompatible bitstream if the descriptions are sent to the decoder, or an increase in complexity at the decoder to generate these descriptions. Our modifications are done only at the encoder, and produce a standard compatible bitstream. Note also that, in Fig. 9, the curve for “next 20 frames” is almost always above the curve for “next 50 frames”. This suggests that, as we move away from the LTR frames, the percentage of MBs using the LTR frame decreases. So, the effect of the LTR frame fades with time.

To decide if the current frame should be designated an LTR frame, we calculate the activity of the current frame to be encoded with respect to the current LTR frame. If the number of active MBs is more than some adaptive threshold (denoted by

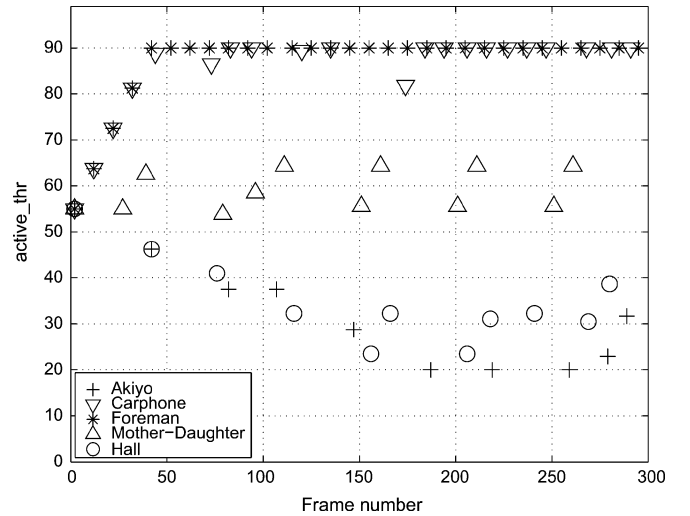


Fig. 10. LTR frame locations and *active_thr* for various video streams. Frame number 2 is the first LTR frame with *active_thr* of 55.

active_thr), then it is time for the next LTR frame. Depending on the frequency of LTR frames, we update *active_thr*. The *active_thr* is incremented by *thr_incr* amount, given by

$$thr_incr = \frac{1}{4} \times (avg_ltr_dist - curr_ltr_dist) \times \frac{(max_mot_thr - min_mot_thr)}{(max_ltr_dist - min_ltr_dist)} \quad (10)$$

where *min_ltr_dist* is set to 10 frames which is our chosen threshold for the minimum number of frames for which an LTR frame is useful (without a scene change). A smaller number of frames can also be considered for *min_ltr_dist* for high-motion videos but this may result in too frequent LTRs to assign higher quality. *max_ltr_dist* is set to 40 frames which is our chosen threshold for the maximum number of frames for which an LTR frame is useful. *avg_ltr_dist* is set to 25 frames (average of *min_ltr_dist* and *max_ltr_dist*) between two LTRs, similar to the one used in [23]. These values are determined experimentally using a large set of videos. The *curr_ltr_dist* is the distance between the current frame and the last assigned LTR frame. We increase *active_thr* to space out the future LTR frames, so as not to use up the bit budget, if the LTR frames have been assigned frequently. If the LTR frames are assigned far apart then *thr_incr* is negative which decreases *active_thr* to reduce the LTR distance.

We do not take future frames into account for choosing LTR locations. Therefore, this method will fail to determine good LTR locations in some cases such as a scene change. In such a case, ideally the new LTR frame should be assigned at the scene change. If the scene change location is less than the minimum LTR distance, then the algorithm will assign the next LTR soon after the minimum LTR distance, and the LTR frame will again be useful for subsequent frames. If we use future frames then the LTR frame selection performance will increase but at the expense of large delay.

Fig. 10 shows the result of our LTR frame selection method along with the change in *active_thr* with the frame number. The simulation was performed for 300 frames with the second

frame as the starting LTR frame and an initial *active_thr* of 55, which is the average number of active MBs per frame when averaged over a large set of QCIF videos containing both high- and low-motion videos. The choice of initial value of *active_thr* minimally affects the overall performance since its value is automatically adjusted depending on the motion of the video stream. The x axis represents the frame number and the y axis represents the *active_thr*. Each set of symbols represents a different video stream and the marked symbols show the locations of LTR frames and its corresponding *active_thr*. For low-motion streams such as Akiyo, we see that *active_thr* tends to decrease from the beginning and the LTR frames are usually separated by a large distance. In contrast, for videos like Foreman which are relatively higher motion streams, the LTR frames occur frequently and *active_thr* quickly saturates to the maximum limit. This LTR frame selection process improves video quality compared to evenly spaced LTR frames for nearly all of the video streams.

D. Multiplexing Video Streams Using Dual-Frame Video Coding With Unevenly Spaced High-Quality LTR Frames

The quality of multiple videos using eLTR_EB and eLTR_ES can further be improved by choosing the high-quality LTR frames based on their motion activity. The first high-quality LTR frame in each video stream is assigned sequentially. For the first video, the high-quality LTR frame is the first P-frame. The high-quality LTR frame for the next video is assigned one frame after the LTR of the previous video. After the first LTR frame, the location of the next LTR frame for any video stream will be calculated using the activity detection algorithm as discussed earlier. Since the location of the next LTR frame is unknown at the time of encoding the current LTR, we do not know the number of regular frames between the two LTR frames that will be used to extract the extra bits for the LTR frames. Therefore, we assume that the next LTR frame will be assigned after the same number of frames as the distance between the previous two LTR frames. We then extract the extra bits evenly from these frames for the LTR frame. Depending on the actual location of the next LTR frame, we calculate the excess (or shortage) of bits that were previously allocated. These bits are distributed among the regular frames between the next two LTR frames. This multiplexing method is called **LTR_EB**.

Similarly, we can also improve eLTR_ES by incorporating motion based high-quality LTR frame selection. Again, the bit allocation for LTR frames is as described in eLTR_EB. This method uses improved dual-frame video coding with high-quality LTR frames and equal slope allocation for regular frames and is denoted by **LTR_ES**. The depiction of LTR_EB is the same as eLTR_EB which is shown in Fig. 7(a) and the depiction of LTR_ES is the same as eLTR_ES which is shown in Fig. 7(b). We expect LTR_ES will perform better than LTR_EB because, in LTR_ES, bits are allocated to different video streams based on their complexity. We expect that LTR_EB will perform better than eLTR_EB, and LTR_ES will perform better than eLTR_ES, because of the advantages of adaptive location of high-quality LTR frames.

IV. MULTIPLEXING VIDEO STREAMS WITH A DELAY CONSTRAINED RATE CONTROL

The methods described earlier did not use rate control, and the MSE improvement over STR_EB is achieved only by comparing the relative complexity across video streams at each frame and by using dual-frame video coding with high-quality LTR frames. The performance of all these methods can further be improved by using rate control which also exploits the relative complexity across the frames in each video. The quality improvement due to rate control comes with a penalty of increased delay at the encoder. With a fixed delay constraint, the encoder may drop a frame partially and this may cause severe error propagation. In this section, we propose multiplexing methods using rate control with a fixed delay buffer. First we describe delay components at the encoder and then discuss how to improve rate control with a fixed encoding delay for various multiplexing methods.

A. Encoding Delay and Rate Control

Delay at the encoder comes from input buffer delay, encoder processing delay, and output buffer delay as shown in Figs. 1 and 2. We use I-P-P coding format where the frames are either Intra (I)-coded or Inter (P)-coded. The I-frames are encoded independently while the P-frames use only previously encoded frames for reference. Consequently all frames are processed sequentially. Therefore, there is a constant input buffer delay of one frame for any video stream. If we use the Bidirectional (B)-coded frames where the B-coded frames use both past and future frames for referencing, then we need to buffer the future frames in order to encode the B-frames and that will increase the size of the input delay buffer. The processing delay is platform dependent and, for the purpose of rate control, we ignore this delay. The encoder generates a variable size of encoded bit-stream for each frame while we assume transmission at a constant bit-rate. Therefore, we need to store bits in an encoder output buffer. The size of the output buffer controls the tightness of the rate control algorithm. For an end-to-end delay in video transmission, we also need to take into account the propagation delay and the delays at the decoder which are the same as the delays at the encoder but in the reverse order.

In practical scenarios, frames are assigned bits based on their relative complexities. Frame complexity is often estimated using Mean Absolute Difference (MAD) which is the difference between the original frame and the predicted frame. Since the current frame is not yet encoded, the H.264 rate control [25] algorithm predicts the current frame MAD from the previously encoded frame MAD. More accurate MAD prediction algorithms such as [26] are proposed in the literature. Given the MAD and a target bit-rate for the current frame, the QP is calculated using a quadratic R-D model. Since complex frames use more bits, an encoder output buffer is needed. We use the terms encoder output buffer and delay buffer interchangeably. The frames (or part of a frame) that exceed the buffer limit are dropped. This leads to error propagation and video quality deterioration.

In the following sections, we first consider multiplexing methods using rate control for dual-frame video coding with

separate delay buffers. We then extend the multiplexing methods to consider a joint delay buffer.

B. Rate Control for Multiplexing Using Dual-Frame Video Coding With Separate Delay Buffers

We again start with STR_EB as the most simple method of bit allocation among multiple video streams. This method neither shares bits among the videos nor shares bits across frames of a single video stream. The basic idea of rate control is to share bits across frames of a video stream in such a way that high motion or complex frames get more bits. By doing so, we increase the overall quality of a video stream. For a fixed delay buffer, we apply the H.264 rate control where the encoder uses the R-D optimization to efficiently encode each video stream separately. We denote this method **STR_RC**. To avoid buffer overflow, we set a Buffer Fullness Threshold (BFT). We start adjusting (increasing) the QP if the buffer occupancy exceeds the BFT which is set to 50%. Otherwise, we let the H.264 R-D optimization determine the QP. If the encoded frame size exceeds the available buffer size, then we drop MBs using the skip mode. The skipped MBs are reconstructed using motion compensated prediction from the STR frame where neighboring motion vectors are used to estimate the motion vector of the lost MB.

With the addition of high-quality LTR frames in dual-frame video coding where many bits are assigned to the LTR frames, the chances of a portion of an LTR frame getting dropped is higher than for a regular frame. This also happens in the rate control implementation given in [27] where two separate rate control paths, one each for the regular and LTR frames, were used by the encoder. In [27], the bit-rate for LTR frames was assumed to be three times that of the regular frames. The rate control implementation in H.264/AVC was used for both the paths and quality improvement was shown over video coding with two STR frames. Since the LTR frames are encoded with separate rate control, there may be cases where the quality of an LTR and its adjacent regular frames are similar, thus losing the importance of an LTR frame. The method of rate control for dual-frame coding in [27] uses the skip mode to drop MBs in case a frame would cause a buffer overflow. While this rate control method works well for high bit-rates, the LTR frames in low bit-rate coding suffer many MB drops due to their large size.

In our approach, we again set a BFT for rate control using a delay buffer for encoding each video stream separately. Most of the time, BFT is set to bf_low , some predetermined fraction of total buffer size. If an LTR frame comes, then we increase BFT to a higher level (bf_high) because we know that LTR frames are assigned more bits compared to other frames. After the LTR frame, we slowly reduce BFT from bf_high to bf_low within bf_slope frames. This process is shown in Fig. 11. We use the H.264 rate control if the buffer fullness is below BFT, otherwise we increase the QP. Note that bf_high , bf_low , and bf_slope were determined experimentally using a set of training videos and were not optimized for any particular type of video.

With this single rate control path to accommodate both regular and LTR frames, the LTR frames are of higher quality than adjacent frames yet we seldom need to skip MBs to avoid buffer

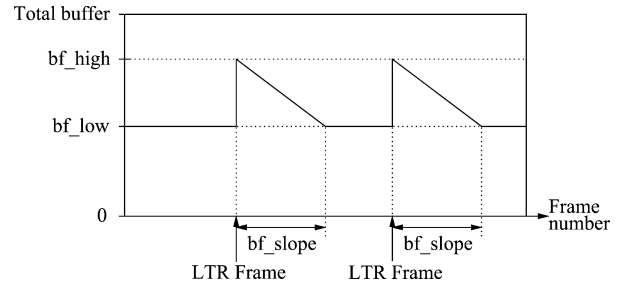


Fig. 11. Buffer fullness threshold (BFT) for the proposed rate control algorithm to encode individual video streams using dual-frame video coding with high-quality LTR frames.

overflow. The number of bits for a high-quality LTR frame is determined by motion activity as discussed in Section III-A.

The detailed discussion about buffer constrained rate control for dual-frame video coding with high-quality LTR frames is given in [16]. We denote this rate control approach for dual-frame video coding with evenly spaced high-quality LTR frames as **eLTR_RC**. The quality is improved over dual-frame video coding with 2 STR frames, and over 1 STR and 1 LTR frame in [27] by our modified rate control algorithm.

The performance of eLTR_RC can further be improved by selecting the locations of the high-quality LTR frames, in addition to their quality levels, using motion activity as given in Section III-C. We denote this method **LTR_RC**. The buffer control operates as in the case for eLTR_RC. The quality improvement in LTR_RC over eLTR_RC is attributed only to the adaptive location of high-quality LTR frames.

C. Rate Control for Multiplexing Using Dual-Frame Video Coding With a Joint Delay Buffer

In multiplexing video streams, we replace the separate encoder output buffer for each video stream by one common encoder output buffer as shown in Fig. 2, and, therefore, the above multiplexing methods need to be modified for one common delay buffer. The rate control extension of STR_ES is denoted by **STR_ES_RC**. Here, the recommended rate control [25] for the H.264 reference software was used to predict the combined frame level complexity of all the videos in order to determine the target bit-rate. Given the combined target bit-rate, the equal slope technique was then applied to allocate bits among videos. Both the rate control and equal slope technique improve the quality when compared to the STR_EB method. Similarly, **SF_RC** is the rate control extension of SF_EB. The target bit-rate for the superframe was assigned by calculating the relative superframe complexity and then superframes are encoded with H.264 rate control (bit constraint for individual superframes is waived). Quality improvement over SF_EB is achieved by assigning the bits to a superframe based on their relative complexity.

While there are many coding variations that can improve compression performance, the use of high-quality LTR frames has an intuitive appeal in a delay buffer constrained multiplexing scenario. Because the LTR frames which demand extra buffer space can be staggered among the different

multiplexed video streams, all videos get some benefit from high-quality LTR coding, avoiding overflowing the buffer. With high-quality evenly spaced LTR frames in dual-frame video coding, (eLTR_RC), each encoder independently allocates the bits to its frames based on the relative complexity and its delay buffer. In eLTR_ES_RC, the multiplexing method not only assigns the bits to a frame based on the relative frame complexity in a video stream but also considers the complexity among the video streams at the frame level with a joint delay buffer, thus improving the overall quality. LTR_ES_RC is a modified multiplexing method of eLTR_ES_RC where the LTR frames are selected using motion activity. We still use the same method to predict the MAD to estimate the complexity of each frame. The QP for a frame in each video is decided using equal slope based on the MAD prediction and target bit-rate.

With multiple video streams having LTR frames at different locations, we need to make sure that the combined buffer fullness should not cross the BFT. Bits for LTR frames are assigned based on the method described in Section III-A. Following each LTR, we decrease the buffer fullness in bf_slope frames. The QP is increased for a frame in any video if it causes the buffer fullness to go above BFT. With a common buffer, the LTR frames are not limited by the small buffer size and it is possible to take more advantage of LTR frames in such a case. The QP for each video is adjusted in such a way that the total bits for all the videos at any time should not overflow the output buffer. As in the previous case, MBs are skipped to avoid buffer overflow. The importance of a combined output buffer for various multiplexing methods will become more clear in the simulation results.

For multiplexing video streams using dual-frame video coding, we assign the high-quality LTR frames using the method described above. For a delay buffer, it is sometimes difficult to accommodate the high-quality LTR frames for different videos close to each other. Suppose there are two video streams to be multiplexed together and both the streams are about to assign an LTR frame close to each other. Suppose we are currently encoding frame $k - 1$. We calculate the motion between the current LTR frame and frame $k - 1$ for both the videos. We also know the motion between the current LTR frame and frame $k - 2$ for both the videos. By extrapolating these two motions, suppose we predict that both the videos will exceed their *active_thr* for frame k . Then the LTR frame of a video that is moving faster towards its *active_thr* compared to the other video is moved ahead by one frame to frame $k - 1$ (as long as frame $k - 1$ is not within the minimum LTR distance). The LTR frame of the other video which is moving slower towards its *active_thr* is delayed by one frame to frame $k + 1$ (as long as frame $k + 1$ is not beyond the maximum LTR distance). By doing so, we create some space between the LTR frames and allocate the desired number of bits to each LTR frame, yet avoid overflowing the buffer.

In summary, we have the following multiplexing methods using delay constrained rate control.

- 1) **STR_RC**: This is the rate control extension of STR_EB. A target bit-rate was assigned and the encoder was allowed

to use R-D optimization to efficiently encode each video stream separately for a given size of its separate output buffer.

- 2) **STR_ES_RC**: This is the rate control extension of STR_ES. Here the recommended rate control for the H.264 reference software was used to predict the combined frame level complexity of all the videos and then the equal slope technique was applied to allocate bits among videos.
- 3) **eLTR_RC**: This is the rate control extension of eLTR_EB. Each video stream is encoded separately with dual-frame coding with high-quality LTR frames. The target bit-rate for a frame is calculated using the frame complexity. Bits are taken from the regular frames and given to the LTR frames. Quality improvement over STR_EB is achieved using rate control and dual-frame coding with evenly spaced high-quality LTR frames.
- 4) **eLTR_ES_RC**: This is the rate control extension of eLTR_ES. First the combined frame level complexity is estimated as described for eLTR_RC and then the equal slope technique is applied. This method combines rate control with equal slope and dual-frame coding with high-quality LTR frames to further improve the overall video quality over STR_EB.
- 5) **LTR_RC**: This method is the rate control extension of LTR_EB. It is similar to eLTR_RC with the exception that the locations of high-quality LTR frames are chosen using the motion activity of a video.
- 6) **LTR_ES_RC**: This is the rate control extension of LTR_ES. It is similar to eLTR_ES_RC with the exception that the locations of high-quality LTR frames are chosen using the motion activity of a video.
- 7) **SF_RC**: This is the rate control extension of SF_EB. The target bit-rate for the superframe was assigned by calculating the relative superframe complexity and then superframes are encoded with H.264 rate control. Quality improvement over SF_EB is achieved by assigning the bits to a superframe based on their relative complexity.

For LTR_ES_RC, the number of bits for a frame in a video is estimated by predicting the complexity of that frame. With the total bits at each frame level, we apply the equal slope technique to all the frames except the LTR frames. Thus, we combine the LTR_ES with rate control to achieve the bit allocation for different video streams. With rate control, the above seven methods should perform better than their respective methods without rate control. We expect that LTR_ES_RC will perform better than the other multiplexing methods discussed in this paper because it has the triple advantage of high-quality LTR frames, rate control, and equal slope bit allocation among videos.

V. RESULTS

The video multiplexing methods described in the previous sections were simulated using the baseline profile of H.264/AVC [28]. The H.264/AVC reference software JM 10.1 [19] was modified for our simulation purpose. All the video streams used for the simulations are QCIF (176×144 pixels)

TABLE I
MSE AND PSNR FOR MULTIPLEXING TWO VIDEO STREAMS

	STR _EB	STR _ES	SF _EB	eLTR _EB	eLTR _ES	LTR _EB	LTR _ES	STR _RC	STR _ES_RC	SF _RC	eLTR _RC	eLTR _ES_RC	LTR _RC	LTR _ES_RC
Foreman (MSE)	114.5	71.2	65.2	107.2	78.8	101.0	75.5	108.6	67.0	61.7	110.9	74.0	104.1	68.7
Foreman (PSNR)	27.54	29.60	29.99	27.83	29.17	28.09	29.35	27.77	29.87	30.23	27.68	29.44	27.95	29.76
Akiyo (MSE)	20.8	29.2	37.6	10.6	14.6	10.7	14.0	9.2	26.6	34.4	7.7	14.7	8.1	14.9
Akiyo (PSNR)	34.95	33.47	32.38	37.89	36.50	37.85	36.67	38.50	33.89	32.76	39.25	36.47	39.06	36.40
Average MSE	67.7	50.2	51.4	58.9	46.7	55.8	44.7	58.9	46.8	48.1	59.3	44.3	56.1	41.8
PSNR (Avg MSE)	29.83	31.12	31.02	30.43	31.44	30.66	31.62	30.43	31.43	31.31	30.40	31.66	30.64	31.92

at 30 frames per second and of length 300 frames. The first frame is an I-frame and the remaining frames are P-frames. The multiplexing methods can be used for any Group Of Picture (GOP) structure. We can either use the equal slope technique for I-frames if the GOP is of the same size for all the videos, or the I-frames can be encoded traditionally where its QP is determined by the QPs from the previous GOP. We can still use our multiplexing methods if there are B-frames with the exception that the B-frames that are not used for referencing can not be used as LTR frames. We considered a lossless channel at various bit-rates for the simulation.

Table I shows the results of multiplexing two video streams at 60 kbps. For any video and any multiplexing method in Table I, the MSE is averaged within and across all the frames. The average MSE for any multiplexing method shows the MSE averaged over both the video streams. The PSNR for any video stream is calculated from the overall MSE of that video stream, and the average PSNR is calculated from the MSE averaged over all frames of both video streams. The following inferences can be drawn from the table above.

- For higher motion videos such as Foreman, STR_ES reduces its MSE by assigning more bits. Lower motion videos such as Akiyo receive fewer bits and experience an increase in MSE. The MSE reduction for the high-motion video is much larger than the MSE increase for the low motion video when compared to the STR_EB case. Therefore, there is a large reduction in overall MSE.
- eLTR_EB uses evenly spaced high-quality LTR frames. We see MSE reduction in both the videos compared to STR_EB. This shows the advantage of using LTR frames and assigning appropriate high quality to them. In this case, both the videos receive equal numbers of bits. The lower motion video, Akiyo, gains more from the high-quality LTR frames than the higher motion video.
- When the equal slope technique is applied along with evenly spaced LTR frames in eLTR_ES, we see that Foreman further reduces its MSE compared to eLTR_EB while there is some increase in MSE for Akiyo. Again, because of equal slope allocation, Foreman receives more bits than Akiyo.
- Although both STR_ES and SF_EB assign bits to each video stream based on the complexity, the performance of STR_ES is better than that of SF_EB due to the fact that STR_ES gives the optimal bit allocation based on the R-D curve of each video stream [(2) and (3)] at the frame level.
- Comparing evenly spaced high-quality LTR frames in dual-frame video coding with the high-quality LTR location found using the activity detection algorithm, we

see that our method of finding LTR location, in general, performs better than taking evenly spaced LTR frames. Therefore, LTR_ES and LTR_EB perform better than eLTR_ES and eLTR_EB, respectively.

Of all the multiplexing methods without rate control, we see that LTR_ES performs the best. When we compare the overall MSE, we find that LTR_ES performs better than STR_EB by 1.79 dB, STR_ES by 0.50 dB, SF_EB by 0.60 dB, eLTR_EB by 1.19 dB, eLTR_ES by 0.18 dB, and LTR_EB by 0.96 dB.

During simulation trials, it was observed that if we allocate many bits to LTR frames, then the performance of LTR_EB tends to be very close to LTR_ES. When more bits are given to LTR frames, then fewer bits are left for other frames. Therefore, when we equalize the slope for these other frames, there are not many bits to adjust and we get a very small advantage from doing equal slope. If we give fewer bits to LTR frames, then the effect of LTR frames is small. In that case, the performance of LTR_EB is close to that of STR_EB. Therefore, it is necessary to moderate the amount of extra quality given to LTR frames to achieve better performance.

Table I also shows the results for the multiplexing methods using rate control. In general, the following trends can be observed from the table.

- Using rate control, STR_RC performs better than STR_EB by 0.60 dB because, with an output buffer, there is more freedom in assigning the bits to various frames according to their relative complexity.
- The multiplexing methods using rate control, STR_ES_RC, LTR_RC, LTR_ES_RC and SF_RC perform better than their counterparts without rate control: STR_ES, LTR_EB, LTR_ES and SF_EB, respectively.
- Using the equal slope technique, STR_ES_RC marginally outperforms SF_RC.
- With the help of dual-frame video coding, LTR_RC performs better than STR_RC and LTR_ES_RC outperforms STR_ES_RC.

Overall, LTR_ES_RC outperforms STR_RC by 1.49 dB, LTR_RC by 1.28 dB, SF_RC by 0.61 dB, and STR_ES_RC by 0.49 dB. When comparing individual video streams we find that, for high-motion video streams, equal slope allocation reduces the MSE by a huge margin, but at the expense of an increase in the MSE for low motion video streams. On the other hand, dual-frame video coding decreases the MSE by a small amount for high-motion videos but it is more successful in reducing the MSE for low motion videos. Rate control also reduces the MSE of each video stream separately. The combination of dual-frame video coding, equal slope allocation,

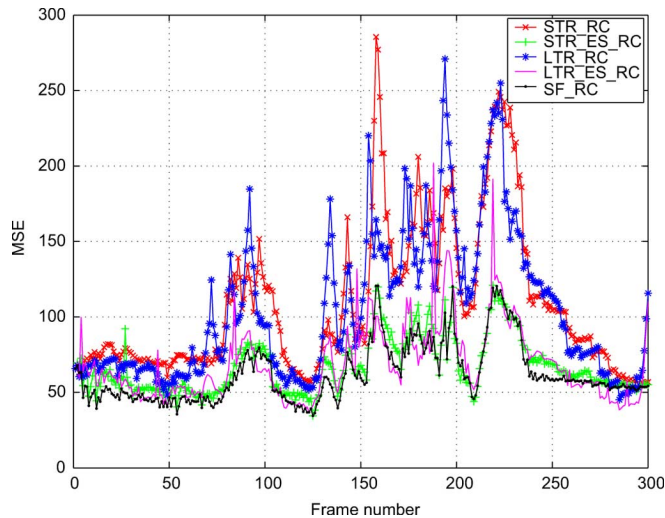


Fig. 12. MSE variation with frame number for Foreman in multiplexed video streams.

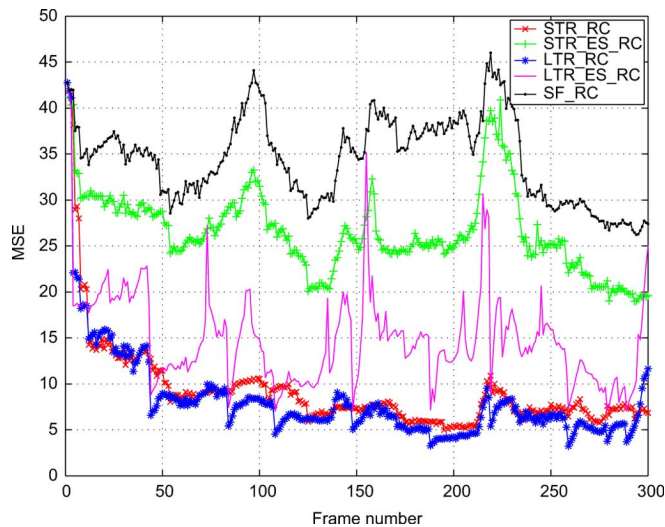


Fig. 13. MSE variation with frame number for Akiyo in multiplexed video streams.

and rate control outperforms all other methods of multiplexing multiple video streams.

Figs. 12 and 13 show the MSE versus frame number for both the multiplexed video streams. The five curves in each figure represent different methods of multiplexing video streams with rate control. For clarity, we only plot five methods for multiplexing that involve rate control and LTR frame selection. The curve at the bottom represents the lowest (best) MSE and the curve at the top represents the highest (worst) MSE. As can be seen, SF_RC squeezes the Akiyo video and gives many bits to the Foreman video, producing a large MSE increase when compared with STR_RC. The STR_ES_RC performs close to SF_RC in Foreman but it produces much better results for Akiyo. The MSE variation of LTR_RC is similar to STR_RC for Foreman, meaning that the dual-frame video coding does not improve the result by a large amount individually. Even though the difference between STR_RC and LTR_RC for

Akiyo is not clear from the figure, LTR_RC produces a large PSNR increase compared to STR_RC. The performance of LTR_ES_RC is close to SF_RC for Foreman but it does much better than SF_RC in Akiyo. The effect of LTR frames is not clearly visible in the figure for Foreman due to frequent small increments in the LTR frame quality but the MSE drops in Akiyo show the presence of high-quality LTR frames. Since the entire video is of high-quality, the quality fluctuations are not perceptually noticeable when viewing the video, but overall the high-quality LTR frames increase the quality of the entire video stream.

A similar result is shown in Table II where four video streams are multiplexed together at a combined bit-rate of 120 kbps. Carphone and Coastguard are of higher motion than Grandma and Akiyo. Note that the MSE and its corresponding PSNR for Akiyo is slightly different in Tables I and II for the cases where the video streams are encoded separately using dual-frame video coding (eLTR_EB, LTR_EB, eLTR_RC, LTR_RC). Depending on the number of video streams to be multiplexed, the starting LTR frame number is different. The remaining LTR frames are dependent on the starting LTR frame number. So, the overall performance is slightly different. Methods involving equal slope improve the performance of Carphone and Coastguard at the expense of Grandma and Akiyo. Use of LTR frames improves the quality of all the videos. Again, all rate control methods perform better than the corresponding ones without rate control. The performance of multiplexing methods improves by finding the LTR location using the motion activity detection. STR_RC performs worst among all the rate control methods and LTR_ES_RC performs the best and is about 0.77 dB better than STR_RC. In this case, the performances of SF_RC and STR_ES_RC are quite similar. Although we expect STR_ES_RC to outperform SF_RC because STR_ES_RC uses equal slope allocation which would be optimal on a per frame basis if the R-D curves are continuous, the actual performances are quite similar due to the discrete R-D curve, which means STR_ES_RC chooses an operating point that is close to but usually not equal to the optimal one. In general, video quality is improved by using dual-frame video coding with equal slope allocation and rate control.

The observed MSE reduction will be less if videos with similar motion levels are multiplexed together. In such a case, all the videos will gain by using dual-frame video coding and rate control. But the advantage of using equal slope allocation will be limited since the complexity of the videos is similar, so there will not be a huge MSE reduction for one video at the expense of a small MSE increase for some other video. For very high-motion videos, even the dual-frame video coding method with high-quality LTR frames fails to achieve large MSE reductions.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed and compared various methods for allocating bit-rate for multiple video streams using dual-frame video coding. We considered the three scenarios of (a) no delay constraint, (b) separate encoder output buffer constraints, and (c) a joint delay buffer for all the multiplexed video streams. The main contributions of this paper are as follows.

TABLE II
MSE AND PSNR FOR MULTIPLEXING FOUR VIDEO STREAMS

	STR _EB	STR _ES	SF _EB	eLTR _EB	eLTR _ES	LTR _EB	LTR _ES	STR _RC	STR _ES_RC	SF _RC	eLTR _RC	eLTR _ES_RC	LTR _RC	LTR _ES_RC
Carphone (MSE)	85.9	61.5	56.3	81.0	63.4	78.8	61.7	81.1	58.0	52.8	81.7	61.7	81.3	60.6
Carphone (PSNR)	28.79	30.24	30.63	29.05	30.11	29.16	30.23	29.04	30.49	30.90	29.01	30.23	29.03	30.31
Grandma (MSE)	26.3	38.8	47.7	16.8	21.2	17.5	21.3	16.9	35.6	44.2	14.5	20.9	14.6	21.0
Grandma (PSNR)	33.93	32.24	31.35	35.87	34.87	35.71	34.85	35.86	32.62	31.68	36.51	34.92	36.49	34.91
Coastguard (MSE)	144.8	99.6	93.0	141.2	108.6	134.7	103.9	125.0	96.5	86.4	124.3	104.4	126.2	98.8
Coastguard (PSNR)	26.52	28.15	28.45	26.63	27.77	26.84	27.96	27.16	28.29	28.76	27.19	27.95	27.12	28.18
Akiyo (MSE)	20.8	30.4	39.8	10.4	13.5	10.8	13.8	9.2	27.8	34.9	7.4	13.4	7.4	13.7
Akiyo (PSNR)	34.95	33.31	32.13	37.98	36.83	37.81	36.74	38.50	33.68	32.70	39.43	36.85	39.43	36.77
Average MSE	69.4	57.6	59.2	62.3	51.7	60.4	50.2	58.0	54.5	54.6	57.0	50.1	57.4	48.5
PSNR (Avg MSE)	29.71	30.53	30.41	30.18	31.00	30.32	31.13	30.50	30.77	30.76	30.57	31.13	30.54	31.27

1) First, separate from the multiplexing problem, we have made three contributions towards improved dual-frame coding.

- The number of bits, and, therefore, the quality level, to be assigned to an LTR frame can be determined using a simple video activity measure, and this performs better than the previous work which allocated a fixed number of bits to the high-quality LTR frames.
- By using the activity measurement algorithm to detect when the LTR frame is becoming obsolete, we developed a simple algorithm for adaptive selection of LTR frame location, and this was shown to reduce the MSE of almost all videos compared to evenly spaced LTRs.
- High-quality LTR frames require more bits on average than regular frames, and the standard approaches for buffer-constrained rate control are not designed for this. We designed a rate control algorithm that uses a tighter target buffer level for most frames, and a less restrictive buffer fullness threshold at and after an LTR frame, and this approach outperformed previous methods.

2) Second, with regard to the multiplexing problem, we have made two main contributions.

- The consideration of R-D properties for performing bit-rate allocation by the equal slope technique is well established. This technique allocates more bits to a video that is going through high motion by taking bits from low motion videos, resulting in a large MSE reduction for high-motion videos with a small increase in MSE for low motion videos. Our contribution was to combine this approach with dual-frame coding with high-quality LTR frames, in which the LTR frames are allocated bits based on motion activity, and other frames are allocated bits using the equal slope technique.
- The buffer-constrained rate control method which works well for individual video streams was modified for a multiplexing scenario, in that the LTR frames in various streams can be slightly delayed or advanced in their locations in order to avoid having them occur at the same time, thereby overflowing the buffer.

In summary, we proposed multiplexing video streams using the triple advantages of (a) dual-frame coding with high-quality LTR frames, (b) modified rate control which accommodates high-quality LTR frames, and (c) equal slope bit allocation modified to accommodate high-quality LTR frames. This new method was shown to outperform existing methods for

multiplexing video streams, including the superframe method equipped with rate control.

There are various avenues for future work. One could use the motion vectors which are computed anyway as part of the encoding process to determine the location and quality level of the LTR frames. With a sufficient number of future frames in the buffer, a dynamic programming solution or a greedy approach can be used to obtain the LTR frame location. The performance of such methods may depend on the lookahead buffer size. Generation of R-D curves involves huge computational complexity. Further research needs to be done to reduce this complexity in order to use such multiplexing methods for practical purposes. In the current work, we assumed a lossless channel. Another avenue for future work involves cross-layer design for transmission of multiple video streams on a wireless channel. In such a case, the decision for the LTR frame location and quality may also depend on the channel conditions.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their many constructive comments.

REFERENCES

- [1] Z. Chen and K. Ngan, "Recent advances in rate control for video coding," *Signal Process.: Image Commun.*, vol. 22, no. 1, pp. 19–38, Jan. 2007.
- [2] X. Zhu, E. Setton, and B. Girod, "Rate allocation for multi-camera surveillance over an ad hoc wireless network," in *Proc. Picture Coding Symp.*, Dec. 2004, pp. 7–12.
- [3] X. Zhu and B. Girod, "Distributed rate allocation for multi-stream video transmission over ad-hoc networks," in *Proc. IEEE Int. Conf. Image Processing*, Sep. 2005, vol. 2, pp. 157–160.
- [4] J. Kammin and M. Sakurai, "Video multiplexing for the MPEG-2 VBR encoder using a deterministic method," in *Proc. IEEE Int. Conf. Automated Production of Cross Media Content for Multi-Channel Distribution*, Dec. 2006, pp. 221–228.
- [5] L. Wang and A. Vincent, "Bit allocation and constraints for joint coding of multiple video programs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 9, pp. 949–959, Sep. 1999.
- [6] J. Yang, X. Fang, and H. Xiong, "A joint rate control scheme for H.264 encoding of multiple video sequences," *IEEE Trans. Consum. Electron.*, vol. 51, no. 5, pp. 617–623, May 2005.
- [7] G. Su and M. Wu, "Efficient bandwidth resource allocation for low-delay multiuser video streaming," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, pp. 1124–1137, Sep. 2005.
- [8] M. Tagliasacchi, G. Valenzise, and S. Tubaro, "Minimum variance optimal rate allocation for multiplexed H.264/AVC bitstreams," *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1129–1143, Jul. 2008.
- [9] L. Boroczky, A. Ngai, and E. Westermann, "Statistical multiplexing using MPEG-2 video encoders," *IBM J. Res. Develop.*, vol. 43, no. 4, pp. 511–520, Jul. 1999.

- [10] M. Kalman and B. Girod, "Optimal channel-time allocation for the transmission of multiple video streams over a shared channel," in *Proc. IEEE Workshop Multimedia Signal Processing*, Oct. 2005, pp. 1–4.
- [11] T. Fukuhara, K. Asai, and T. Murakami, "Very low bit-rate video coding with block partitioning and adaptive selection of two time-differential frame memories," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 2, pp. 212–220, Feb. 1997.
- [12] T. Wiegand, X. Zhang, and B. Girod, "Long-term memory motion-compensated prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 2, pp. 70–84, Feb. 1999.
- [13] V. Chellappa, P. Cosman, and G. Voelker, "Dual-frame motion compensation with uneven quality assignment," in *Proc. IEEE Data Compression Conf.*, Mar. 2004, pp. 262–271.
- [14] M. Tiwari and P. Cosman, "Dual-frame video coding with pulsed quality and a lookahead buffer," in *Proc. IEEE Data Compression Conf.*, Mar. 2006, pp. 372–381.
- [15] M. Tiwari, T. Groves, and P. Cosman, "Multiplexing video streams using dual-frame video coding," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Mar. 2008, pp. 693–696.
- [16] M. Tiwari, T. Groves, and P. Cosman, "Buffer constrained rate control for low bitrate dual-frame video coding," in *Proc. IEEE Int. Conf. Image Processing*, Oct. 2008, pp. 2484–2487.
- [17] M. Gothe and J. Vaisey, "Improving motion compensation using multiple temporal frames," in *Proc. IEEE Pacific Rim Conf. Communications, Computers and Signal Processing*, May 1993, vol. 1, pp. 157–160.
- [18] M. Budagavi and J. D. Gibson, "Multiframe video coding for improved performance over wireless channels," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 252–265, Feb. 2001.
- [19] H.264/AVC ref. Software [Online]. Available: <http://iphome.hhi.de/suehring/tml>
- [20] K. Stuhlmüller, N. Färber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE J. Sel. Areas Commun.*, vol. 18, pp. 1012–1032, Jun. 2000.
- [21] L. Lin and A. Ortega, "Bit-rate control using piecewise approximated rate-distortion characteristics," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 446–459, Aug. 1998.
- [22] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [23] M. Tiwari and P. Cosman, "Selection of long-term reference frames in dual-frame video coding using simulated annealing," *IEEE Signal Process. Lett.*, vol. 15, pp. 249–252, 2008.
- [24] J. Ruiz-Hidalgo and P. Salembier, "On the use of indexing metadata to improve the efficiency of video compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 410–419, Mar. 2006.
- [25] K.-P. Lim, G. Sullivan, and T. Wiegand, "Text description of joint model reference encoding methods and decoding concealment methods," *JVT of ISO/IEC MPEG and ITU-T VCEG, JVT-K049*, Mar. 2004.
- [26] M. Jiang and N. Ling, "Low-delay rate control for real-time H.264/AVC video coding," *IEEE Trans. Multimedia*, vol. 9, no. 6, pp. 467–477, Jun. 2006.
- [27] A. Leontaris and P. C. Cosman, "Compression efficiency and delay tradeoffs for hierarchical B-pictures and pulsed-quality frames," *IEEE Trans. Image Process.*, vol. 16, no. 7, pp. 1726–1740, Jul. 2007.
- [28] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.



Mayank Tiwari (S'03) received the B.E. degree in electronics and communication engineering from the Indian Institute of Technology, Roorkee, India, in 1999, and the M.S. degree in electrical engineering (communication and signal processing) from the Arizona State University in 2004. He is currently pursuing the Ph.D. degree in electrical and computer engineering (signal and image processing) at the University of California at San Diego, La Jolla.

From June 1999 to May 2001, he was with Hughes Software Systems in Gurgaon, India, working on the design and development of video and speech codecs. From June 2001 to August 2002, he was with Cybernetics Infotech Inc. in Rockville, Maryland, working on the design and development of low bitrate speech codecs. His research interests include image and video compression, and mobile multimedia communication.



Theodore Groves received the B.A. degree from Harvard University, Cambridge, MA, and the Ph.D. degree in economics from the University of California, Berkeley, in 1970.

Prior to coming to the University of California at San Diego (UCSD), La Jolla, as a Professor of economics in 1979, he was a faculty member at the University of Wisconsin, Madison, Northwestern University's Kellogg School of Management, and Stanford University, Stanford, CA. He was a founder of mechanism design theory and the discoverer of the "Groves Mechanism" for eliciting truthful information in an incentive-compatible manner. He and coauthor J. Ledyard also developed the first general solution to the "free rider problem" of public goods. He has also studied the Chinese economy's transition to a market economy, optimal policies for minimizing the occurrence of oil spills, and, currently, improved methods for the multiplexing of videos over wireless transmission channels. He is the Director of the Center for Environmental Economics in the Department of Economics at UCSD and is involved in research on water pricing, consumer responses to smart-meter technology for electrical energy consumption, and numerous projects for managing marine resources and the protection of endangered species.

Dr. Groves is an elected Fellow of the Econometric Society and the American Academy of Arts and Sciences.



Pamela C. Cosman (S'88–M'93–SM'00–F'08) 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 during 1993–1995. In 1995, she joined the faculty of the Department of Electrical and Computer Engineering, University of California at

San Diego, La Jolla, where she is currently a Professor. She was the Director of the Center for Wireless Communications from 2006 to 2008. Her research interests are in the areas of image and video compression and processing and wireless communications.

Dr. Cosman is the recipient of the ECE Departmental Graduate Teaching Award (1996), a Career Award from the National Science Foundation (1996–1999), a Powell Faculty Fellowship (1997–1998), and a Globecom 2008 Best Paper Award. She was a guest editor of the June 2000 special issue of the *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS* on "Error-resilient image and video coding," and was the Technical Program Chair of the 1998 Information Theory Workshop in San Diego. She was an associate editor of the *IEEE COMMUNICATIONS LETTERS* (1998–2001), and an associate editor of the *IEEE SIGNAL PROCESSING LETTERS* (2001–2005). She was the Editor-in-Chief (2006–2009) as well as Senior Editor (2003–2005, 2010–present) of the *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*. She is a member of Tau Beta Pi and Sigma Xi.