LOW-LATENCY METHODS FOR WIRELESS VIDEO TRANSMISSION

Alan S. Del Tredici, Joel Rosiene, and Truong Q. Nguyen

University of California at San Diego
Department of Electrical and Computer Engineering
9500 Gilman Drive, La Jolla, CA 92093-0407
http://videoprocessing.ucsd.edu

ABSTRACT

This paper describes a technique for delivering real-time video over low-bandwidth wireless channels. The proposed approach transmits data at a fixed bit-rate specified by the encoder and provides an error correction mechanism for robust data transmission without the use of channel quality information. The performance of this technique in the presence of transmission errors is demonstrated.

1. INTRODUCTION

Predictive coding approaches such as motion compensation are well-known techniques for compressing digital video data [1]. Such techniques by themselves are not well suited to wireless channels, however. Wireless channels can experience transmission errors due to fading, shadowing and environmental noise. These errors corrupt the data being transmitted and can lead to the complete loss of network packets. While such losses will affect any video data stream, these losses are more serious in the case of predictive video coding because the prediction loop propagates errors.

Over the past few years, researchers have proposed a variety of approaches to increase the robustness of low bit-rate video communications [2, 3, 4]. These approaches fail, however, to combine robust transmission with low latency and low complexity required by real-time communication systems. The technique presented here is akin to cyclic and adaptive intra refresh applied to a wavelet based coder [5, 6, 7].

The encoding scheme proposed here addresses the needs of real-time wireless video communication. To minimize latency, each encoded frame contains a fixed number of bits. Within the constraint of these fixed-size frames we have developed an encoding method that recovers from transmission errors. Low-complexity modules are used to construct the encoder and decoder, making the system suitable for real-time applications.

The authors may be reached via e-mail at *adeltred@ucsd.edu, jrosiene@ece.ucsd.edu, nguyent@ece.ucsd.edu.* This work is supported by the von Liebig Foundation at UCSD.

2. ENCODER DESCRIPTION

Figure 1 is a block diagram of the motion-compensating encoder used as the basis for this project. The encoder consists of motion estimation and motion compensation modules, a domain transformation module, and an entropy encoder. Each of these modules is realized with a low-complexity implementation, easing the overall computational requirements of the codec.

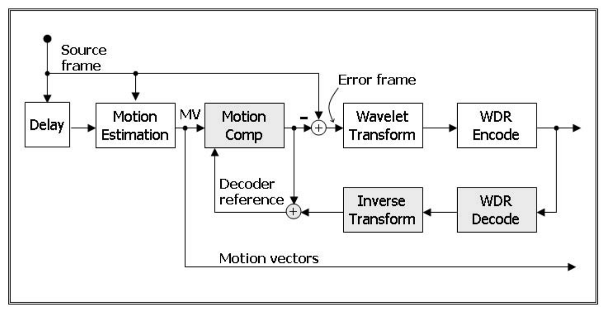

Fig. 1. Encoder block diagram. The shaded blocks are those modules that are also used in the decoder

The encoder performs motion estimation using the phase correlation technique [8]. The domain transformation we use is a wavelet transform [9]. This transform allows a high rate of compression to be achieved by the entropy encoder stage when encoding natural images, and can be implemented entirely with integer computations. The Wavelet Difference Reduction (WDR) entropy encoder, which is also integer-based, produces encoded output with a specified number of bits [10]. The ability to exactly specify the size of the encoder output allows us to guarantee a fixed bit rate.

Image coding is performed with a block-based approach. To generate an inter-coded frame (i.e., a predicted or *P-frame*), image blocks from the current source image are compared with the previous source image to generate motion vectors – one vector per block. These motion vectors are transmitted across the channel to a decoder. They are also used to motion-compensate the current reference image, which is the same image being maintained as a reference at the decoder. From this motion-compensated image, an *error frame* is generated by computing the pixel-by-pixel difference between the source image and the motion-compensated image. The error frame is transformed into the wavelet domain and then entropy coded; the latter encoding step includes the data quantization which makes this compression scheme inherently lossy.

To generate intra-coded or *I-frame* data, the source image is sent directly to the wavelet transform stage; this is equivalent to generating an error frame using a reference image composed of all zeros. The transform and encoding steps are the same for both I and P frames.

The encoder performs all the operations necessary to decode the image as well: WDR decoding and inverse wavelet transformation of the error frame, and combination of resulting error frame with the motion-compensated reference frame. These steps are performed in order to recreate the reference maintained by the decoder. In this way, the motion vectors generated by the encoder are created with respect to the reference image available at the decoder. The encoder, however, receives no confirmation of the correctness of this reference image from the decoder; it is implicit in the algorithm that the encoder and decoder have access to the same reference data.

2.1. Application Model

To focus our efforts towards solutions suitable for wireless video communication, we chose to use the following system requirements as guideline in the development of the proposed encoding approach:

1. Latency: The latency between encoding the source image and displaying the resulting decoding image to be less than 10 milliseconds. Delay greater than 10 milliseconds in an interactive video environment is deemed unsuitable.

2. Image Size: We transmit 8-bit gray-scale SQCIF images (128 x 96 pixels). This is an image size suitable for display on a mobile telephone.

3. Bandwidth: The target channel bandwidth is 48000 bits per second (bps). This bit-rate allows for easy expansion to color images at 64000 bps (using a subsampled color space), which is currently the bandwidth allocated to a video teleconferencing channel.

4. Frame rate: We are targeting a frame rate of 20 frames per second. Full motion video is typically 30 frames per second, while 15 frames per second is generally adequate for teleconferencing applications.

5. The transmission channel is uni-directional and lossy: Due to the small size of our encoded frames (2400 bits), each frame is sent across the wireless network as an individual packet. Transmission errors that are not corrected by the network protocol result in loss of the entire packet. Therefore we model all uncorrectable transmission errors as lost packets.

2.2. Interleaving

The most intuitive approach to correcting transmission errors is to insert I-frames into the video stream at regular intervals. Since I-frames do not take advantage of temporal redundancies, an encoded I-frame requires significantly more bits than a P-frame to achieve the same visual quality. Inserting these I-frames increases the latency of the video stream and requires additional data buffering at the decoder. If an I-frame is encoded with fewer bits in order to reduced latency, the resulting low-quality I-frame creates visual anomalies of its own.

To provide an updated reference image on a continual basis while minimizing the visual artifacts that occur when intra-coded information is inserted into the video sequence, we propose creating a new type of motion compensation frame, the *interleaved frame*. We divide frames into N regions: an interleaved frame contains intra-coded information for one region, and inter-coded information for the remaining regions. With each successive interleaved frame, a different region is intra-coded. After N frames, each region has been intra-coded once, at which point the cycle is repeated.

We call the intra- and inter-coded portions of an interleaved frame the *I-segment* and *P-segment*, respectively. Although the total number of bits within an interleaved frame is constant, the number of bits allocated to the intra- and inter-coded portions can be adjusted.

After evaluating encoder performance using various segment configurations, we chose to divide the SQCIF image into 16 vertical columns, creating segment regions 8 pixels wide and 96 pixels high. This configuration produces I-segments that are small enough to be intra-coded within the bit-budget limit for the frame, yet large enough to exhibit the spatial redundancies that provide data compression via the wavelet transformation. With $N = 16$ regions and a frame rate of 20 frames / second, we are able to refresh the entire frame with intra-coded data in less than one second.

2.3. I and P-Segment Bit Allocation Examples

We next consider how to determine the appropriate allocation of bits between the I and P segments within an encoded frame packet. Poor allocation degrades the quality of the encoded image. As an extreme example, Figure 2 shows a frame taken from an encoding of the Akiyo sequence where each I-segment is allocated 10% of the total bit budget for the frame. In this situation, the I-segment is poorly reconstructed, resulting in an obvious visual anomaly.

Fig. 2. Encoding with I-bits:P-bits = 10:90

At the other extreme, Figure 3 shows the same image with 90% of the bit budget allocated to the I-segment. There are not enough bits to effectively update the motion in the P-segment, resulting in sharp discontinuity between segments.

Fig. 3. Encoding with I-bits:P-bits = 90:10

Figure 4 shows the same frame using interleaved encoding in a sequence for which the I-segment bit budget was dynamically determined. In this frame, the I-segment was allocated 43% of the bit budget.

2.4. Dynamic Bit Allocation

The key aspect of Figure 4 is consistent image quality between the I-segment and P-segment of that frame. We use PSNR as a measure of the image quality relative to the original image source. For each interleaved frame, the encoder attempts to allocate bits to the I and P-segments such that the measured quality values of reconstructed segments match. If the I-segment quality is lower than the P-segment quality, more bits are allocated to the I-segment in the next frame, and vice versa. When determining bit allocation based on image quality, an important assumption we make is that any reduction of quality in either the I-segment or P-segment is due entirely to the quantization effects of the WDR encoder. Under this assumption, increasing the bit allocation of a segment can only increase (or not affect) the resulting image quality of that segment.

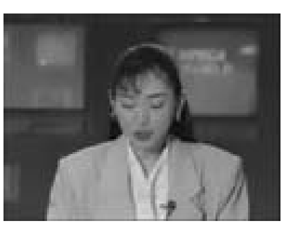

Fig. 4. Encoding with I-bits:P-bits =43:57

2.5. Comparing Quality Measurements

In order to quickly estimate the quality of the I and P segments, we compute the sum absolute-difference (SAD) of the pixel values in each segment with respect to the original source image, obtaining I_{SAD} and P_{SAD}. As the I and P segments are of different sizes, we divide I_{SAD} and P_{SAD} by the number of pixels in the respective segment, obtaining the SAD-per-pixel for each segment, I_{SPP} and P_{SPP}.

Based on our system model, I_{SPP} and P_{SPP} would be zero if not for the quantization error induced by the WDR encoder. Quantization error is by nature a distortion of low-order bits. This implies that the non-zero SAD values we measure are an accumulation of the deviations of the low-order pixel bits from the desired values. Since the amount of information required to encode a value is proportional to the logarithm of that value, we expect the number of encoding bits needed to remove the quantization error from the I and P segments to be proportional to the logarithm of the error value. We therefore compute the base 2 logarithm of I_{SPP} and P_{SPP} to determine the desire change in allocation of encoded bits.

2.6. Bit Allocation Formula

From the reasoning given above, we use the following formulas to calculate the bit allocation of frame *n*:

$$\text{I-bits}_n = \text{I-bits}_{n-1} + \beta * [log_2(P_{SPP_{n-1}}) - log_2(I_{SPPn-1})] \quad (1)$$

$$\text{P-bits}_n = \text{Frame bit budget} - \text{I-bits}_n \quad (2)$$

where:

I-bits_n	=	current I-segment bit allocation
I-bits_{n-1}	=	previous frame I-segment bit allocation
β	=	feedback loop gain factor
P_{SPPn-1}	=	previous frame P-segment SAD-per-pixel value
I_{SPPn-1}	=	previous frame I-segment SAD-per-pixel value

Equation 1 defines a proportional feedback control loop. Tuning of such control loops requires setting the gain factor to a value that allows quick response to changing input values, without allowing the loop to become unstable. We tested the Akiyo, Foreman, and Grandma sequences over a wide range of gain settings. Based on results from these tests we set the loop gain to 32 for all subsequent performance evaluations.

3. RESULTS

With system parameters thus selected, we tested the interleaved method on the three test sequences (Akiyo, Grandma, and Foreman). Figure 5 compares the PSNR of each frame in the Akiyo sequence using several motion compensation techniques. All techniques use fixed-size frames of 2400 bits / frame (48 kbps at 20 frames/sec.)

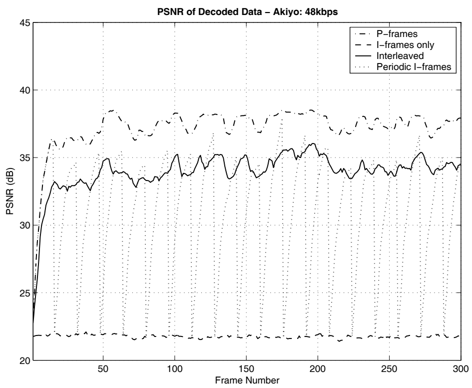

Fig. 5. PSNR for various encoding techniques

The line along the bottom of the graph represents performance using I-frame encoding only. The line with highest PSNR shows performance when continuous P-frames are employed (following an initial I-frame to establish the reference image). As mentioned earlier, this technique is susceptible to degradation if any P-frames are dropped during transmission.

Inserting periodic I-frames to protect against this error mode delivers the spiky performance shown in Figure 5 – each inserted I-frame causes significant degradation of the image quality.

The solid line shows the performance of the interleaving technique. While the PSNR is consistently lower than for the P-frame technique, we have provided error resilience equivalent to the periodic I-frame technique, but without the abrupt image degradation that accompanies I-frames.

Figure 6 shows the effect of dropped frames on the continuous P-frame and interleaved frame sequence PSNRs. For these simulations, we chose (arbitrarily) to drop frames 56, 83, and 126. We see that both the continuous P-frame and interleaved techniques suffer a PSNR reduction from 0.5 dB to 1.0 dB due to a dropped frame. For the continuous P-frame sequences, each dropped frame further reduces the image quality. The interleaved technique is able to return to nominal performance in $N = 16$ frames after a dropped frame, while the P-frame technique has no way of recovering. We obtained similar results for both the Grandma and Foreman test sequences.

4. CONCLUSIONS AND FUTURE WORK

We have demonstrated a method for reliable transmission of compressed digital video over a lossy transmission network

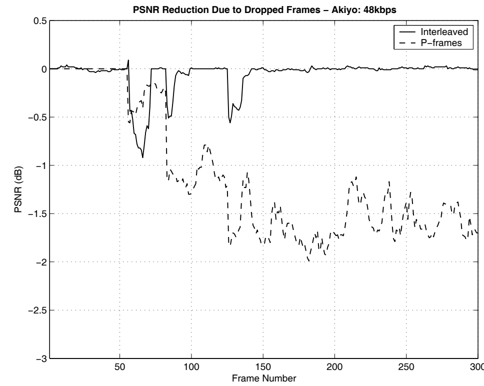

Fig. 6. Effect of Dropped Frames

suitable for a real-time system. The interleaved encoding technique provides the desired resilience to dropped network packets along with low latency which is required for real-time video applications. Further work to understand the relationship of the loop gain factor β to other system parameters would extend the applicability of this approach. Consideration of error conditions other than lost packets may also be of value when working with larger images.

5. REFERENCES

[1] A. M. Tekalp, *Digital Video Processing*, Prentice Hall, 1995.

[2] Qianfu Jiang and S.D. Blostein, "Robust video coding over wireless channesl using TRIRF inter-frame coding," *IEEE Wireless Communications and Networking Conference*, vol. 1, pp. 334–338, Sept. 1999.

[3] J.Y. Liao and J. Villasenor, "Adaptive intra block update for robust transmission of H.263," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 1, pp. 30–35, Feb. 2000.

[4] Y. Wang and Q.F. Zhu, "Error control and concealment for video communication: A review," *Proceedings of the IEEE*, vol. 86, no. 5, May 1998.

[5] K. Imura and Y. Machida, "Error resilient video coding schemes for real-time and low bitrate mobile communications.," *Signal Processing:Image Communication*, vol. 14, pp. 519–530, May 1999.

[6] ISO/IEC 14496-4:2001, *Coding of Visual Objects-Part2:Visual,2d Edition*, 2001.

[7] M. Wollborn, I. Moccagatta, and U. Benzler, "Natural video coding," in *The MPEG-4 book*, F. Pereira and T. Ebrahimi, Eds., IMSC Press Multimedia Series, pp. 345–355. Prentice Hall PTR, 2002.

[8] John Watkinson, *The Art of Digital Video*, Focal Press, 1994.

[9] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, 1996.

[10] J. Tian and R.O. Wells Jr, "Embedded image coding using wavelet-difference-reduction," in *Wavelet Image and Video Compression*, P. Topiwala, Ed., pp. 289–301. Kluwer Academic Publ., Norwell, MA, 1998.