MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Hybrid Distributed Video Coding Using SCA Codes

Emin Martinian, Anthony Vetro, Jonathan Yedidia, Joao Ascenso, Ashish Khisti, Dmitry Malioutov

TR2006-069 October 2006

Abstract

We describe the architecture for our distributed video coding (DVC) system. Some key differences between our work and previous systems include a new method of enabling decoder motion compensation, and the use of serially concatenated accumulate syndrome codes for distributed source coding. To evaluate performance, we compare our system to the H.263+ and H.264/AVC video codecs. Experiments show that our system is comparable to DVC systems from Stanford and Berkeley in the sense that our system performs better than H.263+Intra, but worse than H.263+Inter and H.264/AVC.

Proceedings of the International Workshop on Multimedia Signal Processing (MMSP), October 2006

Hybrid Distributed Video Coding Using SCA Codes

Emin Martinian, Anthony Vetro, and Jonathan S. Yedidia
Mitsubishi Electric Research Labs
Cambridge, MA 02139
Email: {martinian,avetro,yedidia}@merl.com

João Ascenso
Instituto Superior de Engenharia de Lisboa
Lisbon, Portugal
Email: joao.ascenso@lx.it.pt

Ashish Khisti and Dmitry Malioutov
Massachusetts Insitute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139
Email: {khisti,dmm}@mit.edu

***Abstract*— We describe the architecture for our distributed video coding (DVC) system. Some key differences between our work and previous systems include a new method of enabling decoder motion compensation, and the use of serially concatenated accumulate syndrome codes for distributed source coding. To evaluate performance, we compare our system to the H.263+ and H.264/AVC video codecs. Experiments show that our system is comparable to DVC systems from Stanford and Berkeley in the sense that our system performs better than H.263+Intra, but worse than H.263+Inter and H.264/AVC.**

I. Introduction

Distributed video coding (DVC) is a new approach to compression, which shifts complexity from the encoder to the decoder [1], [2]. Current systems achieve this goal by using Slepian-Wolf codes which ignore inter-frame correlation at the encoder, and instead exploit temporal redundancy by doing motion compensation at the decoder. Specifically, [2] uses a combination of Slepian-Wolf decoding and cyclic redundancy check (CRC) codes to help the decoder verify when it has determined the proper motion vectors. In contrast, [1] uses "hash codes" consisting of a few DCT coefficients to help the decoder produce motion vectors.

In this paper, we propose an architecture for DVC based on using Serially Concatenated Accumulate (SCA) syndrome codes for Slepian-Wolf coding and sending a low quality, DPCM/DCT encoded version of the source video to the decoder for use in motion estimation. We refer to this as a "hybrid DVC" system because temporal redundancy is exploited by both the encoder (with lower complexity but lower efficiency) and the decoder (with higher efficiency but higher complexity).

An outline of this paper follows. We describe our system in Section II, and discuss our contributions and its relationship to other methods in III. Next, we present experimental results showing that our system performs better than H.263+Intra, but worse than H.263+Inter and H.264/AVC in Section IV, and close with some concluding remarks in Section V.

II. System Architecture

We begin by describing the overall system architecture for our encoder and decoder illustrated in Fig. 1 and then discussing each component in more detail.

A. Encoder

First, the source video frames are encoded with a low complexity mode of H.264/AVC to produce a low quality reference (LQR), which is sent to the decoder. Specifically, we use the H.264/AVC reference software with a motion search range of zero to produce the LQR, which is sent to the decoder. This so-called "0-motion" mode of AVC achieves better performance than intra-frame coding because it can do inter-frame prediction in a manner analogous to DPCM, but requires far less complexity than inter-frame coding because no motion search is required. As implied by the name, the LQR is of low enough quality that the bit rate is negligible. In particular, the Intra pictures for the LQR are coded at the target PSNR while all non-Intra pictures are coded at a very low quality since the main purpose of the non-Intra components of the LQR are to enable decoder motion estimation described in more detail in Section II-B. Whenever reporting results we count the total rate of the LQR and the distribute video codec.

In parallel with encoding the LQR, we divide each frame of video into 8-by-8 blocks, apply a discrete cosine transform (DCT), divide each transformed value $W[i,j]$ by the corresponding entry from a quantization table[1] $Q[i,j]$, and round $W[i,j]/Q[i,j]$ to the nearest integer. For the quantized high frequency coefficients, we apply run length coding followed by entropy coding similar to JPEG [3]. In particular, we perform a zig-zag scan through the high frequency coefficients, record the number of zero coefficients until the next non-zero coefficient, and then entropy code these run lengths as well as the non-zero coefficient values.

For the low frequency coefficients, we encode the bit planes separately. While most other systems such as H.264/AVC and H.263+, use block based mode selection, working with bit planes has some advantages for DVC systems. First, the correlation with the side information (and hence the efficiency of syndrome coding) depends strongly on the bit plane: if there is a low correlation between the side information and a given bit plane, then entropy coding may be more efficient than syndrome coding. Second, block based mode selection complicates the use of syndrome codes: if block based mode selection were used, then blocks that did not use syndrome coding would reduce the data length and hence decrease the efficiency of the syndrome code and require syndrome codes of varying lengths.

The detailed coding of the bit planes is as follows. First, we organize the low frequency coefficients into bit planes by taking a raster scan of all the 8-by-8 blocks in the frame. For each transform coefficient c, we extract the bth bit plane yielding a set of bit vectors, one for each pair (b,c). Next, for each pair (b,c), a bit plane classifier chooses one of three modes (which is encoded with two bits). The first mode is a skip mode, which is chosen if the bit plane is all zeros. In this case, no further information needs to be sent beyond the mode. The next mode is an entropy coding mode, which is chosen if the binary entropy of the bit plane is less than a fixed threshold. In this case, an entropy coded representation of the bit plane is sent. Finally, if neither of the following conditions applies, a syndrome coding mode is selected and syndromes are generated.

To encode a bit plane vector $\mathbf{v}$ into a syndrome $\mathbf{s}$, we apply the parity check matrix for a serially concatenated accumulate (SCA) code [4] to obtain $\mathbf{s} = \mathbf{H}_0 \cdot \mathbf{v}$. SCA codes have several properties that prove useful for DVC. First, for Gaussian and Laplacian sources, these codes have been shown to perform close to the Slepian-Wolf

[1]Our default quantization table is drawn from the JPEG standard, but adapting $Q[i,j]$ to the sequence improves performance.

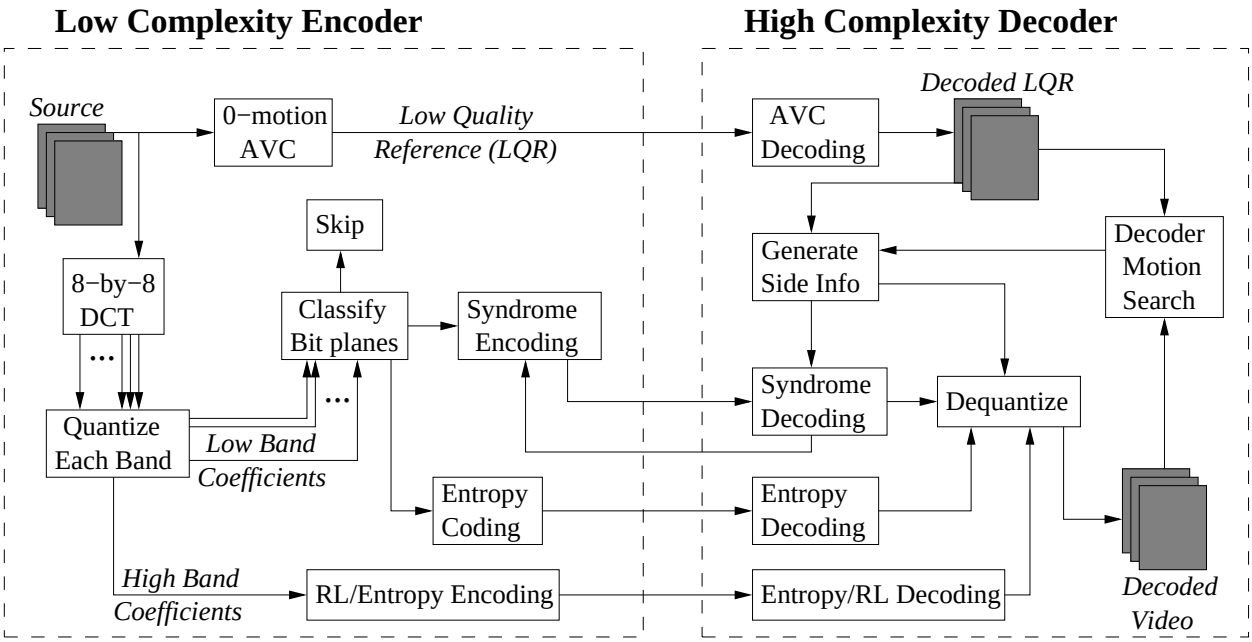

Fig. 1. A diagram of our distributed video coding architecture.

bounds for distributed source coding in [4]. Second, these codes are rate adaptive in the sense that the same basic structure can be used for a wide range of rates. Third, these codes are incremental in the sense that if n_0 bits are sent but determined insufficient for decoding, then $n_1 - n_0$ additional bits can be sent to obtain the same performance as if n_1 bits had been sent initially. Finally, by using belief propagation, these codes can decoded at rates near the Slepian-Wolf limits with complexity proportional to the block length.

B. Decoder

To decode an Intra coded picture at the start/end of a group of pictures (GOP), we decode the corresponding intra picture from the LQR. To describe the decoding process for a non-Intra picture at time t, we assume that frame $t-1$ has already been decoded and saved in a buffer.

In order to estimate motion vectors, the LQR is reconstructed using the H.264/AVC decoder and divided into blocks. For each block in the reconstructed LQR, the motion estimation finds the block in the previously decoded frame that minimizes the sum of absolute differences. Specifically, let $w_{t-1}[i,j]$ denote the value of the pixel at column i and row j of the decoded frame at time $t-1$ and let $\hat{w}_t[i,j]$ denote the corresponding pixel in the LQR. Then the motion vector for the block at column x and row y is

$$\mathbf{v}[x,y] = \arg\min_{a,b} \sum_{i=0}^{7}\sum_{j=0}^{7} \Big| \hat{w}_t[i+8x, j+8y] - w_{t-1}[i+8x-a, j+8y-b] \Big| \quad (1)$$

where a and b take values in the search interval $\{-M, -M-1, \ldots, M\}$ for some integer M.

Once the motion vectors have been determined, the decoder applies them to the previous decoded frame $w_{t-1}[i,j]$ to obtain a motion compensated version of the current frame:

$$\tilde{w}_t[i,j] = w_{t-1}[(i,j) - \mathbf{v}[x(i), y(i)]] \quad (2)$$

where $x(i)$ denotes which 8-by-8 column corresponds to position i and $y(i)$ denotes which 8-by-8 row corresponds to position j.

To generate the log-likelihood ratios required by our belief propagation decoder, we first apply the DCT to each 8-by-8 block of $\tilde{w}_t[i,j]$ to obtain $\tilde{W}_t[i,j]$. Next, we model the probability of the (transformed and quantized) coefficient to be decoded conditioned on the motion compensated current frame using a Laplacian model:

$$p(W_t[i,j] \mid \tilde{W}_t[i,j]) = \frac{\lambda}{2} \exp\left\{-\lambda |W_t[i,j] - \tilde{W}_t[i,j]|\right\}. \quad (3)$$

Although (3) gives the probability of a *coefficient* taking a given value, we need to decode each *bit plane* separately. To obtain the likelihood ratios for a given bit plane, we start with the least significant undecoded bit plane and compute the total probability of a given bit being 0 or 1 by summing over all possible coefficient probabilities where that bit is either 0 or 1. For example, to compute the probability that the least significant bit at position (i,j) is 0, we would compute

$$Pr[\text{lsb of } W_t[i,j] = 0] = \sum_{x=0,2,\ldots} p(W_t[i,j] = x \mid \tilde{W}_t[i,j]). \quad (4)$$

Then, if the least significant bit at that position was determined to be 1, we would compute the probability for the next least significant via

$$Pr[\text{2nd lsb of } W_t[i,j] = 0] = \sum_{x=1,5,\ldots} p(W_t[i,j] = x \mid \tilde{W}_t[i,j]). \quad (5)$$

As first described by Ungerboeck in the context of channel coding [7], syndrome decoding from least to most significant bit planes is better than decoding in the reverse order. Intuitively, this is because once the least significant bit plane has been decoded correctly, knowing the value of the least significant bit plane makes decoding the next bit plane easier since the values to be compared are further apart.

To decode the syndromes and recover the quantized bit planes we apply a turbo-like decoding algorithm based on belief propagation

[4]. If decoding fails, we allow the decoder to request more syndrome bits from the encoder via a feedback channel.[2] Decoding (and further feedback requests) are performed until decoding is successful. In our experimental results, the bit rates reported include the minimum number of syndromes required for all syndromes to be decoded correctly.

III. Discussion

Three key differences in our system compared to other DVC systems [1], [2] are the low quality reference (LQR) produced by the encoder to enable decoder motion estimation, the serially concatenated accumulate syndrome codes to enable rate adaptation and incremental redundancy, and the bit plane based mode selection.

One important advantage of our approach is that the 0-motion mode of H.264/AVC can produce an LQR that provides acceptable decoder motion compensation while requiring very few bits to encode. The LQR method bears more resemblance to the "hash codes" proposed for decoder motion compensation in [1] which consist of a few coarsely quantized low frequency DCT coefficients and less resemblence to the cyclic redundancy checks (CRCs) used for decoder motion compensation proposed in [2]. Nonetheless, an advantage of the LQR over both hash codes and CRCs is that since the LQR is encoded using the sophisticated video coding tools of H.264/AVC, it can be compressed much more efficiently than hash codes or CRCs.

This is especially relevant for sequences with very strong temporal correlation such as the 30 frames/second, CIF resolution "Mother and Daughter" where the LQR can be encoded using much less than 100 kb/s while even sending a one byte CRC or hash code per 8-by-8 block would require about 380 kb/s just for the CRC or hash code! While hash codes (and to a lesser extent CRCs) could also be encoded more efficiently to exploit temporal or spatial correlations, designing a special purpose hash or CRC compression algorithm that performs comparably to H.264/AVC is a non-trivial task.

Another advantage of the LQR is that it provides video information that can enable a variety of decoder motion compensation methods in addition to the block matching approach in (1). For example, other motion estimation tools commonly used at the encoder such as sub-pixel motion estimation and multi-reference prediction can be used at the decoder when the LQR is available. In particular, Li and Delp [5] showed that half-integer motion estimation provided between 0.3 and 0.7 dB improvements in the quality of the side information[3] while multi-reference search yielded improvements between 0.7 and 2.7 dB.

Along these lines, we noted that the "noisy motion estimation" problem in our architecture differs from classical motion estimation in that we try to match the LQR $\hat{w}_t[i,j]$ to the reference in (1) instead of matching the source $w_t[i,j]$ to the reference as would be the case in non-distributed video compression. To account for this we explored regularized motion estimation by including a bias towards small motion vectors and a bias to a smooth motion vector field. The former was implemented by adding a penalty term of the form $\lambda \cdot (a^2 + b^2)$ to (1) while the latter was implemented by adding a penalty term corresponding to the sum of squares of the difference between neighboring motion vectors. This improved the PSNR of the motion compensated side information by up to 1.5 dB for some frames and about 0.4 dB on average with noticeable visual improvement in the quality of the side information. Unforuntately the improvements on the overall rate-distortion performance was signicantly smaller than the 0.4 dB improvement to the quality of the side information. However, these results (and those in [5]) illustrate that having the LQR available enables a wide variety of decoder estimation techniques, which can improve the side information quality.

[2]Of course, in practice a feedback channel might not be available and either some small amount of errors would have to be tolerated or a CRC or other information would have to be provided by the encoder to enable error detection. Nonetheless, this is a common assumption in DVC[1], [2].

[3]Note these are improvements in the quality of the side information not the overall rate-distortion performance.

IV. Experimental Results

To evaluate the performance of our DVC system we encoded the "foreman" (QCIF resolution) and "mother and daughter" (at CIF resolution) sequences and compared the results to various other codecs as illustrated in Fig. 2. In all tests we encoded 100 frames at 30 frames per second. Results for our DVC system include the sum of the bit rate for the DVC system and the LQR. These results show that our DVC system is comparable to similar distributed source coding results [1], [2] in the sense that it outperforms H.263+Intra, but does not perform as well as (or better) than H.263+Inter. This is a promising preliminary result showing that DVC has the potential to improve upon intra-frame coding.

In addition, we note that the gains of our DVC system are larger for the mother and daughter sequence than on foreman. We believe this is because mother and daughter has less motion (and thus much higher temporal correlation) than foreman. Since DVC systems exploit temporal correlation at the decoder we expect the gain of DVC systems over intra coding to be larger on sequences with high temporal correlation. We note, however, that when the temporal correlation is high, one can obtain a low complexity encoder using the 0-motion mode while losing very little relative to full inter coding.

To make DVC practical, however, one should actually compare to the state of the art H.264/AVC codec as well as to other low complexity codecs. Such a comparison is sobering; even the intra mode of H.264/AVC outperforms our DVC system. This is partially due to the fact that H.264/AVC contains much more engineering effort and optimizations than H.263+ and our DVC system. Furthermore, if the goal is to obtain maximum compression efficiency by eliminating the complexity of motion search, one can improve upon the intra mode of H.264/AVC or H.263+ by using inter-frame coding with a motion search range of zero. This so called 0-Motion mode essentially corresponds to differential pulse code modulation (DPCM) and achieves performance very close to inter-frame coding without motion search.

In addition to PSNR results, we also present a visual comparison of the mother and daughter sequence encoded with H.263+Intra and our DVC system in Fig. 3. From these images we see that due to the inefficient compression of intra coding, the H.263+Intra coded sequence suffers from noticeable blocking artifacts. In contrast, since our DVC system can exploit temporal correlation at the decoder, it compresses the sequence more efficiently fewer artifacts (although some ringing is evident).

V. Concluding Remarks

To evaluate the performance of our DVC system, we presented experimental results for the foreman sequence and compared to various modes of the H.264/AVC and H.263+ codecs. As with [1], [2], our system performs better than H.263+Intra, but worse than H.263+Inter. This comparison shows that with roughly equal engineering effort, DVC can provide better compression efficiency with lower encoding complexity than a traditional system. Practically, however, by simply using a motion search range of zero in H.264/AVC or H.263+, one can eliminate the complexity of encoder motion search while achieving

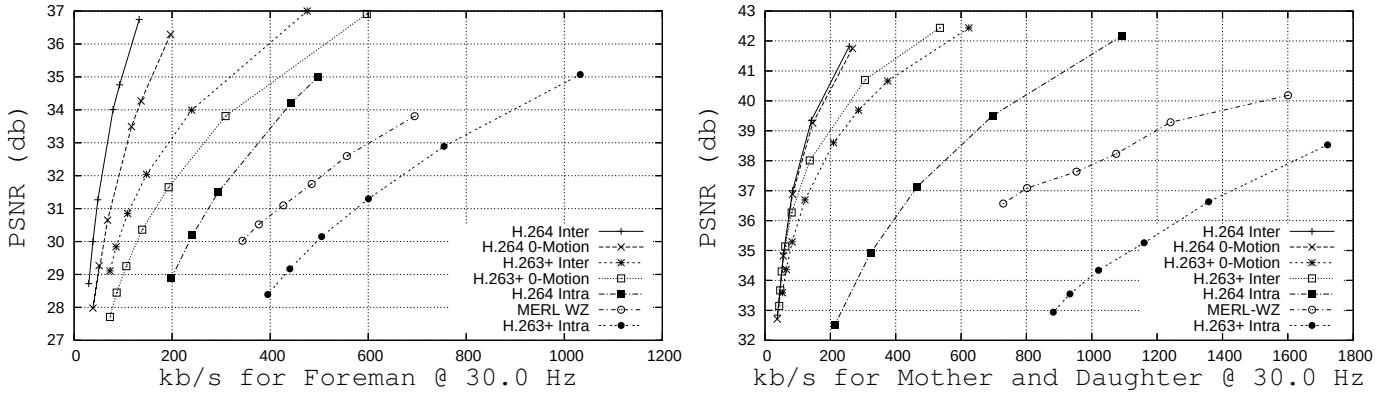

Fig. 2. A comparison of various video coding systems on the first 100 frames of the QCIF resolution foreman sequence and the mother and daughter sequence at 30 frames/second. The "0-motion" modes represent the performance of H.264/AVC or H.263+ with motion vectors forced to be zero. Our distributed video codec is denoted MERL-WZ.

Fig. 3. Frame 14 of the CIF mother and daughter sequence at 1000 kb/second with H.263+Intra (left) and our distributed video codec (right).

close to inter-frame performance. We believe the appropriate conclusion is not that DVC systems are unworkable, but rather that intense research is needed to determine how the weakness of DVC systems can be improved and complemented by the strengths of traditional codecs.

Many options exist for further work. First, since the LQR in our system is already sent, the higher bits in the quantization step are somewhat redundant. An encoder that took into account the LQR when doing transform and quantization (*e.g.*, by only coding the error between the source and the LQR) could potentially improve performance. Ideally, the LQR could efficiently represent low motion areas and the Slepian-Wolf coded bits could efficiently encode high motion areas by exploiting decoder motion estimation. Second, the accuracy of side information correlation models such as (3) leave much to be desired and can be improved by the methods in [8], [6] or by using the decoder motion estimation error between the LQR and the best matching block in (1) to weight the side information.

REFERENCES

[1] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 71–83, January 2005.

[2] R. Puri and K. Ramchandran, "PRISM: an uplink-friendly multimedia coding paradigm," in *International Conference on Acoustics Speech and Signal Processing*, April 2003, vol. 4, pp. IV – 856–9.

[3] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions On Consumer Electronics*, vol. 38, no. 1, pp. xviii – xxxiv, February 1992.

[4] J. Chen, A. Khisti, D. M. Malioutov, and J. S. Yedidia, "Distributed source coding using serially-concatenated-accumulate codes," in *Information Theory Workshop*, October 2004, pp. 209–214.

[5] Zhen Li and Edward J. Delp, "Wyner-Ziv video side estimator: Conventional motion search methods revisited," in *International Conference on Image Processing*, Genoa, Italy, September 2005.

[6] Min Wu, A. Vetro, J. S. Yedidia, Huifang Sun, and Chang Wen Chen, "A study of encoding and decoding techniques for syndrome-based video coding," in *IEEE Symposium on Circuits and Systems*, May 2005, pp. 3527–3530.

[7] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Transactions on Information Theory*, vol. 28, no. 1, pp. 55–67, January 1982.

[8] R.P. Westerlaken, R. Klein Gunnewiek, and R.L. Lagendijk, "The role of the virtual channel in distributed source coding of video," in *International Conference on Image Processing*, September 2005.