SYNDROME-BASED LIGHT-WEIGHT VIDEO CODING FOR MOBILE WIRELESS APPLICATION

Min Wu, Guogang Hua, Chang Wen Chen

Department of Electrical and Computer Engineering
Florida Institute of Technology
Melbourne, FL 32901

ABSTRACT

In conventional video coding, the complexity of an encoder is generally much higher than that of a decoder because of operations such as motion estimation consume significant computational resources. Such codec architecture is suitable for downlink transmission model of broadcast. However, in the contemporary applications of mobile wireless video uplink transmission, it is desirable to have low complexity video encoder to meet the resource limitations on the mobile devices. Recent advances in distributed video source coding provide potential reverse in computational complexity for encoder and decoder [1, 2]. In the same spirit, we proposed in this paper a syndrome-based light-weight video encoding scheme for mobile wireless applications. This scheme is based on two innovations: (1) adoption of low resolution low quality reference frames for motion estimation at the decoder; (2) introduction of more powerful product accumulate code. Extensive experimental results have confirmed that this syndrome based encoding can reduce computational complexity at the encoder while maintaining good reconstruction quality at the decoder. Therefore, this light weight video coding scheme is suitable for mobile wireless applications.

1. INTRODUCTION

In current video coding standards, the complexity of an encoder is generally much higher than that of a decoder because some encoding specific components, such as motion estimation, have computationally intensive operations even when efficient fast motion search is used [3]. Such architecture is suitable for downlink transmission model of TV broadcast when the system has few encoders and numerous decoders. In contemporary media-rich uplink wireless video transmission, for instance, a video camera cell phone transmits video wirelessly to the base station, complexity is of paramount concern because battery-powered mobile handheld devices usually have limited processing power and memory. Therefore, it is desirable to have a low complexity video encoder to meet the resource limitations.

Recently, distributed video coding schemes have been proposed to provide potential reverse in computational complexity for decoder and encoder [1, 2]. The theoretical background of these schemes is based on Slepian-Wolf and Wyner-Ziv distributed source coding theories [4, 5]. The recently proposed video coding schemes built on the distributed source coding principles have a general architecture: the encoder applies error correcting codes to each frame and generate syndrome bits [1, 2]. The decoder estimates each frame. Such estimate can be viewed as a noisy version of the original frame. The error correcting coder combines the syndrome bits and the noisy version to reconstruct each frame. The less the estimation error, the less the syndrome bits are needed.

There are various studies [1, 2, 6, 7] that have been conducted in syndrome-based distributed video coding. In [1], error correcting coding is applied to each block on a frame, in order to not exceed the correcting ability, the coefficients are very coarsely quantized, and refine bits need to be sent after distributed coding. In this case, only 20% bits are distributed coded, the rest of the bits are entropy coded. In [2], each frame estimated by the interpolation of previous frames is not very accurate, this will generate more syndrome bits to correct estimation errors. The achieved rate-distortion results are between the results obtained by running inter mode and intra mode of standard video compression, but still far from optimal.

The number of syndrome bits needed for a successful decoding obviously depends on the estimation error. Therefore, building more accurate estimation for each frame will improve the overall coding efficiency. Many research efforts aimed at building more accurate estimation at the decoder [7]. In [7], hash codewords of the current frame are generated and sent to aid the decoder to estimate more accurately the motion. The hash codewords are coarsely quantized version of a downsampled 8 by 8 image block. To save the bit rate, the distance between two hash codewords of co-located blocks on previous and current frame is calculated to decide whether or not to send the codewords. In our system, we use highly compressed version of each frame as reference to perform motion estimation at the decoder in order to build more accurate estimation [8]. Although we add some cost for compression and transmission of those low quality frames, the overall bit rate can be saved because more syndrome bits will be saved by accurate estimation.

On the other hand, adoption of efficient error correcting code will also improve the overall coding efficiency. Trellis code is first used in the PRISM [1]. More efficient Turbo code and LDPC code are then applied in [7, 9, 10]. One of the common features of those methods is that channel coding methods are applied to source coding. To achieve compression, the rate of the selected channel coding must be larger than 1/2, Otherwise, there will be no compression at all. Especially in video coding, we usually expect high compression ratio, so the channel coding methods used must be high-rate code. Product accumulate code has been proven to have good performance with high code rate [11]. Product accumulate code [11] is a serial concatenation of a single-parity-check-based product code, and a rate-1 recursive convolutional code. It provides similar coding performance to turbo code, but with lower error floor. This has the advantage in correcting very small estimation errors. The related research for product accumulate code used for distributed source coding can be found in [12]. Some technical issues of its application in

---

This research work has been supported by Florida Institute of Technolgy Allen S. Henry Endowment Fund.

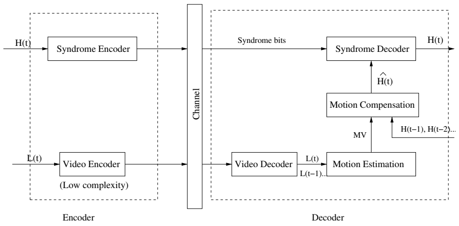


Fig. 1. Architecture of syndrome-based video source codec

distributed video coding are also discussed in [12]. In our system, we adopt product accumulate code in syndrome coding to further improve the overall coding efficiency. Several technical issues are also investigated.

The innovations of our proposed syndrome-based light weight video source coding scheme are two folds. First, we adopt product accumulate code in syndrome coding to improve the coding efficiency. Secondly we adopt highly compressed version of each frame as reference to build more accurate estimation. Extensive experimental results have confirmed that this syndrome based encoding can reduce computational complexity at encoder while maintaining good reconstruction quality at the decoder. Thus this scheme is suitable for mobile wireless applications.

The rest of paper is organized as follows: In Section 2, we describe our system and investigate the compression of low resolution and low quality sequence and the adoption of product accumulate code in distributed video coding. In Section 3, extensive experimental results are given for verification. In section 4, we provide summary with some discussions.

2. SYNDROME-BASED VIDEO SOURCE CODING

Figure 1 shows the architecture of this system. At the encoder, for each frame of the video, two sets of bits are transmitted to the decoder. The first set contains syndrome bits generated in syndrome encoding. The second set contains highly compressed version of the current frame. At the decoder, we use the highly compressed low quality frame as reference to perform motion search and compensation to predict the current high quality frame. The syndrome decoder combines the syndrome bits and the prediction to reconstruct the current high quality frame.

2.1. The Application of Production Accumulate Code

Product accumulate (PA) code has been shown to perform nearly optimally at high rates for channel coding [11]. This makes PA code a promising candidate for syndrome-based distributed video coding, since video sequences have large temporal redundancy, high compression rates are possible. To achieve high compression rates, high-rate channel codes are needed for distributed source coding. To obtain good performance from high-rate punctured convolutional codes and turbo codes, convolutional codes usually must be of long constraint length, making the decoding complexity rather high. LDPCs, on the other hand, provide good performance at possibly lower complexity; however, the encoding complexity can be as high as $O(N^2)$ (N is the codeword length) if direct matrix multiplication is performed. Moreover, explicit storage of a generator matrix may be required [11]. In [11], it was shown that the PA code possess good BER performance, linear encoding/decoding complexity, as well as an efficient soft-decoding algorithm. It performs well even when the code rate is as high as 0.97. Also In [12] PA code is gracefully rate-adaptive. It could use different rate codes with minimal complexity to deal with the changes of the source correlation. PA code is also incremental as shown in [12]. If the decoder needs more bits, encoder can send those additional bits without wasting any information previously sent. All these features of PA code make it a good choice for syndrome-based distributed video coding.

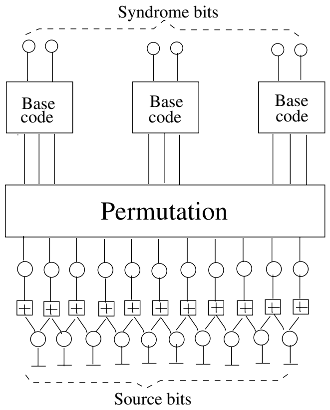


Fig. 2. Syndrome encoder/decoder using PA codes

Figure 2 shows a factor graph for a syndrome-based encoder/decoder using PA codes. The encoding operation starts with the source bits at the bottom of the factor graph, and proceeds deterministically up to the syndrome bits at the top of the factor graph. The decoder uses received syndrome bits and the correlated source to estimate the original message. Before decoding, we first compute the values of any redundant syndrome bits. The decoder then uses a turbo-like algorithm which passes soft extrinsic information between decoders of the base codes and the accumulator code. The accumulator code is decoded using the BCJR algorithm, which generates extrinsic information for each bit for the decoders of the base codes. Side information from the source can be treated just like soft evidence from the channel in channel coding.

2.2. Encoder Side

2.2.1. *Compression of Low Resolution of Low Quality Frames*

We select H.264 AVC to compress the low quality sequences because it has high coding efficiency, comparing with other standards [13]. Full motion search is a very efficiency way for video compression, but with intensive computation load. However, Ting et al observed that motion vectors tends to concentrated in the previous frame [14], no matter the motion is slow or high. This means zero motion vector is a good estimate of full motion search for low quality and low comlexity coding cases. Thus, we use H.264 IPPP zero motion mode to compress the low quality frames.

Compressing down-sampled frames, instead of original frames, can further improve the compression at low bit rate [15]. For those down-sampled frames, more bits per pixel are allocated for each block. Such frame displays more details, and has less blocking artifacts at very low bit rate. Moreover, as we up-sample a frame at decoding side, we can add another improvement to the process since interpolation can further blur the blocking effect.

2.2.2. *Generation of Syndrome Bits*

For each frame of video, the 8x8 integer transform is applied to compact energy at low frequency. Fixed mode of intra prediction is applied to reduce the dynamic range of the coefficients. Which mode to choose is an empirical result by running some extensive simulations. Zero coefficients are coarsely estimated and taken off from the input of syndrome encoder to take the advantage of already known distribution of the coefficients. Product accumulate code is then applied to each bit plane to generate syndrome bits. How many syndrome bits to send can be estimated by how similar the current frame is to the previous frame. Or, it can be preciously decided by the feedback from decoder if additional control channel exists.

2.3. Decoder Side

At the decoder side, we use some information from current low quality frame to improve the prediction. The low quality frames are used to perform motion search to build motion compensation of the current frame. Since successive frames probably have the same blocking effect at very low bit rate, the effect of blocking artifacts could be nullified by the similar blocking artifacts. Therefore, motion search is performed on current low quality frame and previous low quality frames. When the difference between the low and high quality sequences is large, motion compensation is not accurate in prediction. Adding a constraint on motion search for very low quality sequence will improve prediction. The cost function is then:

$$SSD + \lambda(Q, PSNR\_Diff)(MV)^2 \tag{1}$$

where, SSD is sum of squared difference between the blocks in current frame and matching block in previous frame, and MV is motion vectors. λ is an empirical function of the quantization parameter and the PSNR difference between low quality and target high quality sequences.

The syndrome decoder combines the prediction version with syndrome bits to reconstruct the current frame.

3. EXPERIMENTAL RESULTS

In this section, we present the experimental result to confirm the effectiveness of the proposed syndrome-based encoding/decoding presented in Section 2.

3.1. Encoder Side

We use Carphone and Foreman sequences in our experiment. The frame rate is 30 fps. First, we test the performance of IPPP zero motion mode. Figure 3 shows the result on Carphone sequence. The result on Foreman is similar to that on Carphone. We observe that at low PSNR range (less than 30), zero motion mode is closer to IPPP full motion search mode. We also observe that zero motion has higher coding efficiency on slow motion sequence than that on high motion sequence. This phenomenon can be explained as follows. In slow motion sequence, the successive pictures are highly correlated in temporal domain. The motion vectors are more condensed around 0. In contrast, there is much less temporal correlation in fast motion sequences. The motion vectors are less condensed around 0.

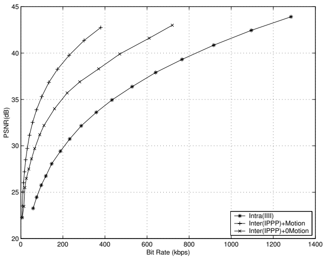


Fig. 3. Zero motion performance on Carphone QCIF

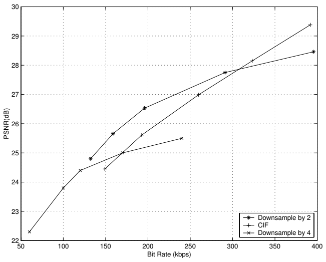


Fig. 4. Performance of downsampling on Foreman CIF

Next, we test the down-sampling approach in low quality sequence compression. Figure 4 show the result on Foreman sequence. The image is compressed with intra (IIII) mode. We test the image down-sampled by factor of 2, and 4. From the result, we observe that when we compress image lower than 24 dB, 4 is the best choice for down-sampling factor; when we compress image between 24 and 28 dB, 2 is the best choice; when we compress image higher than 28 dB, down-sampling provide no benefit in compression.

3.2. Decoder Side

First we test the impact of adaptive constrained motion search on prediction. Figure 5 shows the result on Carphone QCIF sequence. Experimental result shows that the adaptive result is always better than the motion constraint without adaptation.

The PSNR of the reference is selected about 5 dB lower than the target quality. The target quality is set from 30 to 38 dB that is satisfactory with most wireless mobile video broadcasting applications. We test the proposed method on Carphone and Forman frames, respectively. Figure 6a shows the PSNR of reconstructed frames of Carphone sequence. Figure 6b shows the similar results running on Forman sequence. The PSNR of the reconstructed frames is between those results obtained by running inter and intra mode. We also run H.264 with the zero motion mode that can be viewed as the low complexity mode of the standard video encoding. In zero motion mode, the motion search is turned off at the encoder, thus the complexity is

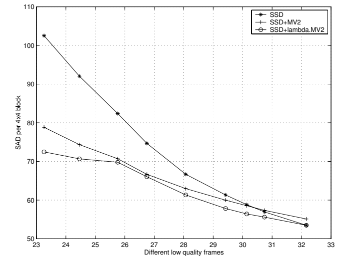
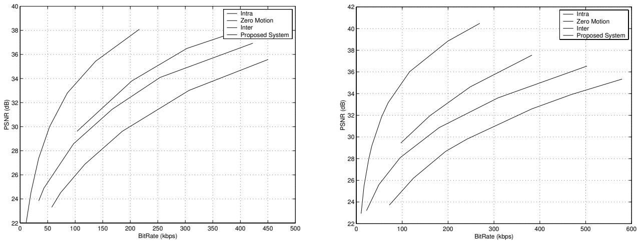


Fig. 5. Impact of adaptive constrained motion search on prediction

comparable to that of the proposed scheme. The PSNR of the reconstructed frames is much higher than the result obtained by running zero motion mode. Comparing with the results in [2, 1], The perfromance of the proposed system is closer to the result obtained by running inter mode.



(a) Carphone

(b) Forman

Fig. 6. PSNR of Reconstructed frames at differenct bitrates

4. SUMMARY AND DISCUSSION

In this paper, we proposed a syndrome-based light-weight video coding scheme for mobile wireless applications. We use very low quality frames as reference to perform motion estimation at the decoder to build more accurate estimate of each frame. In the compression of low resolution and low quality sequence, we conclude that zero motion with down-sampling compression mode is an optimal trade off with processing power constraint. At the decoder, adaptive constrained motion search tends to improve prediction. Furthermore, we adopt more powerful product accumulate code in syndrome coding to allow for the reconstruction of the video at the target quality using fewer syndrome bits. The PSNR of the reconstructed frames of the proposed system is closer to the result obtained by running inter mode. Extensive experimental results have confirmed that this syndrome-based encoding is indeed applicable to mobile wireless applications.

5. REFERENCES

[1] R Puri and K. Ramchandran, “PRISM: A new robust video coding architecture based on distributed compression princples,” in *Proc. of Conference on Communication, Control and Computing*, Allerton, IL, October 2002.

[2] A. Aaron and B. Girod, “Towards practical wyner ziv coding of video,” in *Proc. of IEEE International Conference on Image Processing*, Barcelona, Spain, September 2003.

[3] Y. Liu and S. Oraintara, “Complexity comparison of fast block-matching motion estimation algorithms,” in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 2004, pp. 17–21.

[4] D. Slepian and J. K. Wolf, “Noiseless coding of correlated information sources,” *IEEE Transaction on Information Theory*, vol. 19, pp. 471–490, July 1973.

[5] D. Wyner and J. Ziv, “The rate-distortion function for source coding with side information at the decoder,” *IEEE Transaction on Information Theory*, vol. 22, pp. 1–10, January 1976.

[6] A. Aaron, S. Rane, E. Setton, and B. Girod, “Transform domain wyner-ziv codec for video,” in *in Proc. SPIE Visual Communications and Image Processing*, San Jose, CA, January 2004.

[7] A. Aaron, S. Rane, and B. Girod, “Wyner-ziv video coding with hash-based motion compensation at the receiver,” in *in Proc. IEEE International Conference on Image Processing*, Singapore, October 2004.

[8] M. Wu, A. Vetro, J.S. Yedidia, H. Sun, and C.W. Chen, “A study of encoding and decoding techniques for syndrome-based video coding,” in *IEEE International Sysposium on Circuits and Systems*, Kobe, Japan, May 2005, vol. 4, pp. 3527–3530.

[9] M. Sartipi and F. Fekri, “Distributed source coding in wireless sensor networks using ldpc codes: a non-uniform framework,” in *Proc. IEEE Data Compression Conference 2005*, Mar. 2005, p. 477.

[10] A.D. Liveris, Xiong Z, and C. Georghiades, “Compression of binary sources with side information at the decoder using ldpc codes,” *IEEE Communications Letters*, vol. 6, pp. 440–442, Oct 2002.

[11] J. Li, K.R. Narayanan, and C.N. Georghiades, “Product accumulate codes: a class of codes with near-capacity performance and low decoding complexity,” *IEEE Transactions on Information Theory*, vol. 50, pp. 31–46, January 2004.

[12] J. Chen, A. Khisti, D.M. Malioutov, and J.S. Yedidia, “Distributed source coding using serially-concatenated-accumulate codes,” in *Proc. IEEE Information Theory Workshop 2004*, Oct. 2004, pp. 209–214.

[13] H. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, “Low-complexity transform and quantization in h.264/AVC,” *IEEE Transaction on Circuits and System for Video Technology*, vol. 13, pp. 598–603, July 2003.

[14] C. W. Ting, L. M. Po, and C. H. Cheung, “Center-biased frame selection algorithms for fast multi-frame motion estimation in h.264,” in *Proc. of IEEE International Conference on Neural Networks and Signal Processing*, Nanjing, China, December 2003, pp. 1258–1261.

[15] A. M. Bruckstein, M. Elad, and R. Kimmel, “Down-scaling for better transform compression,” *IEEE Transactions on Image Processing*, vol. 12, pp. 1132–1144, September 2003.