Bitrate Control of Distributed Source Coding by Frame Prediction Method

Kin Honn Chiam and Mohd Fadzli Mohd Salleh
Pusat Pengajian Kejuruteraan Elektrik dan Elektronik Universiti Sains Malaysia, Pulau Pinang 14300, Malaysia

Abstract: Distributed source coding is proposed as a new video compression paradigm to shift the bulk of computation to the decoder, in order to achieve low complexity encoder design. During the decoding operation, the decoder will iteratively request for syndrome bits from the encoder to achieve certain decoding quality. This causes inconsistent bitrate and might increase the bitrate. In this study, we add a module at the decoder to iteratively predict the next to be decoded frame. Hence, the bitrate can be controlled by eliminating the need to request for more syndrome bits from the encoder. The results show that the proposed model can achieve performance close to the conventional distributed source coding model.

Key words: Group of pictures, Wyner-Ziv, frame interpolation, bidirectional motion oracle, distributed source coding.

1. Introduction

Based on the Slepian-Wolf and WZ (Wyner-Ziv) information theoretic result, DSC (distributed source coding) is a new paradigm for video transmission [1]. In the conventional approach of video transmission, statistical dependencies between frames are exploited at the encoder, making the design of the encoder much more complicated than the decoder. On the other hand, the DSC encodes the frames independently but decodes them jointly, thus shifting the complex calculation algorithm from the encoder to the decoder side.

This makes encoding process of DSC very simple by design. Consecutive frames in the incoming video sequence are first split various groups based on the cumulative motion of all frames in each group crossing a pre-defined threshold. The number of frames in a group is defined as the GOP (group of pictures), with the first frame known as the key frame while the other frames are called the WZ frames. The key frame is intra-coded using the conventional way while the WZ frames are inter-frame encoded, with only the syndrome bits sent across the network.

In contrast, the decoding process of the DSC is relatively more complicated. The received key frame is first decoded and then used to generate the SI (side information). The SI is an estimation of the next to be decoded WZ frame, which is only available at the encoder. To complete the decoding operation, syndrome bits are requested from the encoder via a feedback channel. If the correlation between the SI and the syndrome bits is high, fewer syndrome bits need to be requested from the encoder. However, request for syndrome bits will be iterated till a certain decoding quality is reached. This makes the bitrate uncontrollable and might increase the bitrate as more syndrome bits are transmitted from the encoder.

The main goal of this study is to control and reduce the rate of syndrome bits transmission from the encoder. This is achieved by mean of iteratively improve the generation of SI, an estimate of the WZ frame, by using the combined interpolation method with bidirectional motion oracle. This proposed model still achieves the paradigm of the DSC as the complex computation for the WZ frame estimation is performed at the decoder side. The performance of the proposed model is determined by comparing the RD (rate distortion) curve with the results of other
Bitrate Control of Distributed Source Coding by Frame Prediction Method

publications [1, 2].

This paper is structured as follows: The Section 2 first highlights the basic concepts of DSC and the joint decoding with motion oracle method; In Section 3, the details of the proposed DSC codec model, methodology, and implementation are presented; The results with the C++ codes implementation are discussed in the Section 4; followed by conclusions and further work in the Section 5.

2. Theoretical Background

2.1 Practical Schemes for Distributed Source Coding

The DSC paradigm has been implemented in some practical schemes as presented in other literatures [2-4]. While the earliest works explore the encoding scheme in the pixel domain, some subsequent works consider using the transform domain encoding approach. Different binary codes, like the turbo codes and LDPC (low density parity check) codes, are also used as the channel encoding technique in different models.

In Ref. [2], the authors applied the WZ codec to the pixel values of a video sequence. There are two separate encoders and a joint decoder. Assuming that the size of the GOP is two, the key frames are alternating with the WZ frames. One encoder codes the key frames without any knowledge of the WZ frames and sends them to the decoder. At the decoder, a prediction of the WZ frames, which will be used as SI, is predicted.

Such an approach is extended to the transform domain [3]. The discrete cosine transform enables the codec to exploit the statistical dependencies within a frame. This can achieve better RD performance. The authors in Ref. [3] also presented a syndrome-based framework that employs trellis-coded quantization and trellis channel codes, and successively extended to a video system called PRISM. The idea is to consider every video frame as a different source.

Adaptive video splitter is proposed by the authors in Ref. [4] to intelligently split the incoming video stream into the key frames and WZ frames based on the motion. Hence, the output of the video splitter is GOP of various sizes. The video splitter shall also be capable to detect high motion frames in the video sequence. These high motion frames shall only act as the key frames, making the size of the GOP equal to one. Thus, the frames are only encoded using the conventional intra encoder, bypassing the WZ encoder. This makes the model to act as the intra coder.

2.2 Joint Decoding with Motion Oracle

Fig. 1 depicts a typical video compression system that utilizes motion oracle to reliably decode a source message with only motion vector field of the source message available at the decoder. Consider two consecutive quantized video frames, $X$ and $Y$, related by the forward motion vector field, $M$. Let $X$ be the quantized frame to be encoded while $Y$ be the previously decoded quantized frame available at the decoder.

The LDPC encoder computes the syndrome, $S$ of $X$ and transmits them to the decoder. At the decoder side, the SI, $\psi$ is first determined by applying a probability model to the $Y$. Then, the LDPC decoder determines the initial soft estimate, $\theta$ (statistical estimate of $X$) with the aid of SI, $\psi$. The estimation of soft estimate, $\theta$ and SI, $\psi$ will be refined iteratively via the expectation maximization algorithm.

During the next iteration, the motion oracle module informs the probability model of which pixels of $Y$ should be used for the generation of the refined SI, $\psi$. This is done by maintaining a posteriori probability distribution, $P(X)$ on the forward motion vector field,

![Fig. 1 Coding of X With Respect to Y with motion oracle][2].
by comparing $Y$ with the soft estimate, $\theta$. Eq. (1) shows the posteriori probability distribution of source frame, $X$ based on the soft estimate parameter, $\theta$.

$$P(X) = \prod_{i,j} \theta(i,j, X(i,j))$$ (1)

where, $X(i,j)$ defines the pixel values of the source frame, $X$ at the coordinates $(i,j)$. Hence, the posteriori probability distribution of the forward motion vector field, $P(M)$ is given by Eq. (2).

$$P(M) = P(M|Y,S,\theta)$$ (2)

For each iteration, the LDPC decoder sends the motion estimator of a current soft estimate, $\theta$ in order to refine the distribution of on the forward motion vector field, $M$. In return, the probability model updates the SI, $\psi$ for the LDPC decoder by blending information from the pixels of $Y$ according to the refined distribution on the forward motion vector field, $M$. Based on the updated estimate of SI, $\psi$, the LDPC decoder proceeds to determine the next refined soft estimate, $\theta$ and the whole process is repeated.

### 3. Methodology

The architecture of the proposed DSC model and the detailed operation flow are shown in Fig. 2 and Fig. 3 respectively. In the subsequent sub-sections, detailed operations of the encoder module and decoder module in the model are described.

The LDPC codes are chosen as the channel encoding technique is this work as the codes are notable for the advantages in term of bit error rate performance [5, 6]. For a wide range of source correlation, the LDPC codes also achieve better RD performance [6, 7].

#### 3.1 Encoder Module for Distributed Source Coding

Firstly, the video splitter inserts the key frames in between the WZ frames, creating the GOP. The algorithm used by the video splitter shall not significantly increase the complexity of the design of the encoder. Thus, the simple yet powerful metrics such as the difference of histograms and block variance difference are used to evaluate the motion activity along the video sequence.

The key frames are encoded using the conventional H.264 intra coder [8] and transmitted to the decoder directly. On the other hand, the WZ frames are transformed using a $4 \times 4$ discrete cosine transform, and the coefficients are grouped together according to the standard scan zig-zag order into 16 bands.

Each band is then quantized using the standard JPEG quantization matrix with a quality level of 50. This matrix renders high compression without significant degradation in the visual quality of the frames.
After that, the quantized symbols are deciphered into rows of long messages and fed into the LDPC encoder.

The LDPC encoder consists of a LDPC syndrome-former concatenated with an accumulator. For each block of message, syndrome bits are created using the LDPC codes and modulo-2 accumulated to produce the accumulated syndromes. The encoder stores these accumulated syndromes in an input buffer and transmits them continuously to the decoder in chunks.

3.2 Decoder with Motion Oracle Module

At the decoder, by doing the inverse accumulation operation to the received accumulated syndromes, the syndrome bits are extracted. These syndrome bits are fed into the motion oracle, and used to aid and refine the generation of the SI, $\psi$ in the later process.

At the same time, the received key frames are also intra decoded. The decoded output key frames are used to generate the SI, $\psi$ using the frame interpolation method combined with the motion estimation method. The motion oracle module performs a full search based block motion estimation algorithm to estimate the motion between the decoded key frames and the syndrome bits. The search for the candidate block is exhaustively carried out in all possible positions within the specific range of the forward motion vector field, $M$.

To determine if the decoding operation is successful, syndrome check error is computed. The Hamming distance between the syndrome bits and the generated SI, $\psi$ is calculated. If the Hamming distance is non-zero, the decoder proceeds to the next iteration to further refine the SI, $\psi$. A new soft estimate of the SI, $\psi$ is calculated. Based on this value, the motion oracle module selects which values of the syndrome bits are used for the next full search block motion estimation to generate the refined SI, $\psi$.

Using this approach, the decoder does not need to request for any additional syndrome bit from the
encoder to achieve a certain decoding quality. The WZ frames are iteratively predicted by keeping on refining the quality of the generated SI, $\psi$.

On the other hand, if the Hamming distance between the syndrome bits and the generated SI, $\psi$ is zero, the decoding is declared successful.

The decoded bits are then sent for inverse quantization and reconstruction. Finally, the inverse discrete cosine transform operation is carried out. This operation restores the image into the pixel domain from the transform domain.

4. Simulation Result

The DSC model proposed in Section 3 is completely implemented with the C++ programming language. Both encoding and decoding functions of the LDPC codes are imported from the Matlab library [9] and converted to the C++ programming language.

The proposed model is evaluated with the Foreman monitor QCIF video sequence with 15 Hertz vertical frequency. This video sequence has very high amount of motion activity. We consider 299 frames of the sequence coded at a rate of 30 frames per second. We also set the quantization factor for the key frames, in order to have an average peak to signal noise ratio (PSNR) value, of about 33 decibels. To make the fair comparison with other publications [1, 2], only the performance relative to the luminance component of the WZ frames is considered.

Fig. 4 depicts the simulation results of the proposed model. We consider the results relative to a reference model implemented with RE (rate estimation) as described in another publication [2]. In addition, we also compare the results with the performance of a standard H.264 (AVC) video codec with intra frame coding as presented in Ref. [1]. In that publication, the WZ frames are encoded as “B” frames, predicted from the previous and next frames with bidirectional motion compensation, as used in current video standards.

From the data, it can be seen that the PSNR achieved by the proposed model is around 3 to 5 decibels better than that of the H.263 (AVC) intra coder for a given bitrate. This shows that the DSC paradigm can achieve the good reconstruction of output frames even though this information is not available at the decoder. Furthermore, in the proposed model, a relatively low quantization factor is used to compress the key frames. This results in a higher quality of the generated SI to aid the decoding of the WZ frames.

However, the PSNR achieved by the proposed model is around 0.5 to 2.2 decibels lower than that of the reference model implemented with RE. This shows that more syndrome bits are still needed to be requested from the encoder to aid the decoding of the WZ frames. The LDPC codes are a lossy compression coding technique. Hence, some information has been discarded in the received syndrome bits. The refining of the SI based on this syndrome bits would produce the output with incomplete information as well. This might cause the poorer performance of the proposed model.

Moreover, in the real wireless transmission, the received syndrome bits might be subjected to the noise corruption due to multipath fading channel. Hence, the corrupted syndromes bits might not be suitable to be used to provide the soft estimate in refining the SI. The noise corruption during transmission might be propagated to the output. As a forward error correction technique, the LDPC decoder still needs to request for the retransmission of the
syndrome bits and discard the current received syndrome bits if the corruption level in the current received syndrome bits is too high till the original source message could not be recovered.

Finally, the linear interpolation and motion oracle with bidirectional motion estimation might not be the best method to predict the WZ frames. The linear interpolation algorithm only estimates the missing values based on the mean between two neighboring values. This algorithm might not be suitable if there are a lot of edges in the frames.

5. Conclusions

In this paper, the proposed DSC model is designed such that a motion oracle module is added in the decoder to predict the WZ frames. The SI, which is an estimate of the WZ frames, are iteratively predicted and refined at the decoder to match the WZ frames. This eliminates the need of the decoder to iteratively request for more syndrome bits from the encoder. Hence, the bitrate can be controlled and reduced.

With the DSC paradigm, simulation results show that the proposed model outperforms the conventional H.264 (AVC) intra codec. However, the proposed model still does not achieve the RD performance as the conventional DSC model implemented with RE. Nevertheless, the RD performance of the proposed model is still close to the reference model with the deviation of less than 3 decibels.

To improve the quality of the SI generated at the decoder, other methods could be evaluated. Furthermore, the linear interpolation algorithm might not be suitable to predict the frames with a lot of edges. Hence, the proposed model shall also be verified with other video sequences. This remains further research.

References

[6] Sarah, J. “Introducing Low-Density Parity Check Codes.” Instructions in School of Electrical Engineering and Computer Science, the University of Newcastle, Australia.