CONTENT ADAPTIVE GOP SIZE CONTROL WITH FEEDBACK CHANNEL SUPPRESSION IN DISTRIBUTED VIDEO CODING

Charles Yaacoub[1,2], Joumana Farah[1], Béatrice Pesquet-Popescu[2]

[1]Faculty of Sciences and Computer Engineering, Holy-Spirit University of Kaslik, Jounieh, Lebanon
{charlesyaacoub, joumanafarah} @ usek.edu.lb
[2]Signal and Image Processing Dept. Télécom ParisTech, Paris, France
beatrice.pesquet @ telecom-paristech.fr

ABSTRACT

This paper presents a novel algorithm for content adaptive GOP size control in distributed video coding. The GOP size is dynamically varied along the sequence, depending on motion activity. Automatic mode selection allows the system to switch between H.264 intra-coding and Wyner-Ziv coding modes to optimize the overall performance. Furthermore, the encoder determines a suitable compression ratio for the Wyner-Ziv frames without the need for a feedback channel. Simulation results show significant improvement in the average system performance, compared to fixed GOP Wyner-Ziv and H.264 intra-coding.

Index Terms— Video, distributed compression, GOP size control, return channel, Wyner-Ziv.

1. INTRODUCTION

In the last decade, distributed video coding (DVC) [1-10] has become a topic of interest for the research community, especially for applications requiring simple encoders. Based on Slepian-Wolf [11] and Wyner-Ziv [12] theorems, DVC shifts most of the computation burden that usually resides at the encoder, mainly due to motion estimation in traditional video coding [13], to the decoder side.

In practical DVC systems, a subset of frames, known as key frames, is usually intra-coded using a traditional video encoder. One or more frames following each key frame, known as Wyner-Ziv (WZ) frames, are then compressed by appropriate puncturing of the parity bits at the output of a channel coder. At the receiver, side information (SI) is generated by interpolating previously decoded (key or WZ) frames.

Most DVC systems require a feedback channel to allow flexible rate control and to ensure successful decoding of WZ frames [14], which makes these systems impractical in multiuser real-time applications [2]. On the other hand, in order to successfully decode WZ frames at feasible WZ bitrates, high-quality key frames are required. This can lead to a very high bit rate requirement, which is not possible in limited bandwidth applications. Additionally, when the key frames are too distant apart, the quality of the side information is degraded. As a result, most research on DVC considers a GOP (Group of Pictures) of limited size, often 2 or 3, i.e. each key frame is followed by one or two WZ frames.

Several attempts have been made to reduce the use of the return channel. Artigas and Torres [3] and Morbée et al. [4] proposed techniques that rely on performance tables used by the encoder to predict the compression level of each particular frame. Kubasov et al. proposed in [5] an encoder rate control technique that reduces the use of the feedback channel. Transform domain WZ rate control algorithms were introduced in [6] and [7], for DCT and wavelet-based WZ codecs, respectively. On the other hand, several studies considered increasing the GOP size, but in the presence of a feedback channel. Aaron et al. developed in [8] a practical WZ codec with fixed GOP sizes ranging from 2 to 5. As the GOP size increases, the system's performance decreases. However, lower rates could be reached with greater GOP sizes due to the high bit rate requirements of the key frames. In [9], Ascenso et al. present a content adaptive GOP size selection algorithm, where the number of frames in a GOP is determined dynamically, depending on motion activity.

In this paper, we develop a novel algorithm for dynamically varying the GOP size in a DVC encoder. Based on our previous WZ rate estimation technique presented in [2], a feedback channel is not needed for the decoding of WZ frames. Automatic mode selection allows the system to switch to H.264 intra-coding mode in regions where H.264 outperforms WZ video coding. Furthermore, in contrast with [8] and [9], our algorithm can be easily extended to take into account channel impairments and multiuser scenarios, based on our studies in [2] and [10].

This paper is organized as follows. In Section 2, a detailed description of the proposed GOP size control algorithm is presented. Simulation results are reported and discussed in Section 3, and finally, conclusions are drawn in Section 4.

2. GOP SIZE CONTROL ALGORITHM

In a video sequence, low motion results in highly correlated consecutive frames. The aim of varying the GOP

This work was partly supported by a research grant from the Lebanese National Council for Scientific Research (LNCSR) and was realized within the Franco-Lebanese CEDRE program.

size is to allow the system to better exploit this property, by reducing the number of intra-coded key frames in regions where WZ frames would yield a better rate-distortion (R-D) performance. In high motion areas where intra-coding outperforms WZ coding, the GOP structure is reduced to one (H.264 intra-coded) frame per GOP. This automatic mode selection allows the WZ encoder to make use of H.264 coding efficiency to improve the system's R-D performance.

The WZ codec considered in this study is the one developed in [2], to which we add the new GOP size control technique. To this aim, first let S_{max} represent the maximum allowable GOP size. For each GOP, let R_0 represent the average bitrate assigned for the first frame (intra-coded key frame) in the GOP, and $PSNR_0$ its Peak Signal-to-Noise Ratio (PSNR). S_{max} can be chosen depending on the system's delay constraints. For a certain GOP of size S, let F_0 denote the key frame, F_1, F_2, … , F_{S-1} the WZ frames, and F_S the key frame of the next GOP.

```
procedure R_PSNR_estimations(i, len) {

  If len < 2

     Return                    // End of the recursive function calls

  Else

     a = i, b = i+len          // a and b are the time indices
                               // of the frames used during the
                               // interpolation process.

     d = ⌊(b – a)/2⌋           // time interval from a to the
                               // frame at mid distance
                               // between a and b.

     SI(a+d) = Interpolate(a,b)        // Perform average interpolation
                                       // between frames at time
                                       // indices a and b.

     Q(a+d) = QuantizeFrame(F(a+d))

     R(a+d) = EstimateBitrate(Q(a+d))  // estimate the bitrate as
                                       // explained in Section 2.

     F′(a+d) = Reconstruct(Q(a+d), SI(a+d))     // reconstruct the
                                                // quantized WZ frame
                                                // given the estimated
                                                // side information.

     PSNR(a+d) = ComputePSNR(F′(a+d), F(a+d))   // Compute the PSNR
                                                // of the reconstructed
                                                // frame.

     R_PSNR_estimations(i, d);         // Recursive function call using
                                       // the first half of the GOP.

     R_PSNR_estimations(i+d, len-d)    // Recursive function call using
                                       // the second half of the GOP.

  End If

}
```

Fig. 1. Pseudocode of the recursive procedure R_PSNR_estimations(i,len) used to estimate the rate and PSNR for all the frames in a GOP.

To perform GOP length decision, our proposed algorithm operates as follows:

Initially, set S=1.
While $S \leq S_{max}$ do:

If S==1, go to step 5, otherwise:

***Step1**: Interpolate between F_0 and F_S*

Since motion estimation is not applicable at the encoder, for complexity reasons, average interpolation [1] is used. The interpolated frame serves as a rough estimate of the side information available at the decoder, obtained by motion-compensated interpolation, during the decoding process of the WZ frame $F_{\lfloor S/2 \rfloor}$, located at half-distance between F_0 and F_S.

***Step2**: Estimate the average bit rate $R_{\lfloor S/2 \rfloor}$*

Given the WZ frame $F_{\lfloor S/2 \rfloor}$ and its estimated side information, the encoder first determines its lower compression bound using entropy calculations, taking into account the motion level in this frame, as explained in [2]. The WZ frame's compression rate can therefore be estimated by multiplying the lower bound with a constant T_M, depending on the number *M* of quantization bits per pixel (suitable values for T_M were also determined in [2]). The computation of $R_{\lfloor S/2 \rfloor}$ then becomes straightforward.

***Step3**: Compute $PSNR_{\lfloor S/2 \rfloor}$*

Given the WZ frame $F_{\lfloor S/2 \rfloor}$ and its side information estimate, the encoder can determine an estimate $F'_{\lfloor S/2 \rfloor}$ of the frame that will be obtained at the receiver after WZ decoding, by first quantizing the WZ frame, and then reconstructing an 8-bit version using the available side information. The PSNR is then computed between $F_{\lfloor S/2 \rfloor}$ and $F'_{\lfloor S/2 \rfloor}$.

***Step4**: Repeat steps 1 to 3 until rate and PSNR estimates are obtained for all the frames in the GOP.*

The pseudocode of the recursive procedure used to estimate the rate and PSNR for all the frames in a GOP of size S, starting at time index *i*, is shown in Figure 1. *len* is the time interval between the frame at index *i* and the next frame used during the interpolation process (initially, *len* = S). If the interpolation process (in Step 1) involves a key frame at time index k, the real (or H.264 decoded) frame F_k is used, since it will be available at the decoder. However, if the frame involved in the interpolation process is a previously decoded frame, the estimate F'_k (obtained from Step 3 in a previous iteration) is used instead of F_k, because the former estimates better the frame that will be available at the decoder side, since the latter is not known by the decoder.

Step5: *Estimate the average rate and PSNR obtained with a GOP of size S, respectively defined as*:

$$R_{av}^{S} = \frac{1}{S}\sum_{j=0}^{S-1} R_j \text{ and } PSNR_{av}^{S} = \frac{1}{S}\sum_{j=0}^{S-1} PSNR_j .$$

Step6: *Determine* $\lambda_{av}^{S} = PSNR_{av}^{S} / R_{av}^{S}$

This represents the average PSNR per average unit bit rate estimated for a GOP of size S.

Step7: *Increase the GOP size: S = S + 1*

The best R-D performance is obtained by maximizing the average PSNR per unit bit rate. As a result, the system decides the GOP length L as: $L = \underset{k=1,2,\ldots,S_{max}}{\arg\max}\left(\lambda_{av}^{k}\right)$.

In other words, if L=1, an H.264 I-frame is then transmitted. Otherwise, an H.264 intra-coded key frame is transmitted, followed by L-1 WZ frames. This procedure is repeated at the beginning of every GOP and thus, the GOP length is dynamically varied along the sequence, in order to optimize the overall performance.

3. SIMULATION RESULTS

In our simulations, we consider 400 frames from two different QCIF video sequences (Foreman and Grandmother) with different levels of motion, sampled at a rate of 30 frames per second. Each sequence is first encoded using our WZ codec from [2], with fixed GOP sizes ranging from 1 to 5. H.264 coding is performed using JM FRExt reference software, version 13.2, with baseline profile [13]. The results are then compared with the case where a WZ codec with a dynamically varying GOP size is used, as explained in Section 2, with S_{max} set to 5.

In figures 2 and 3, we show the average R-D curves obtained by averaging the rate and PSNR over all the sequence (key and WZ frames). Different rate points are obtained by varying the quantization parameter M for the WZ frames (M = 1, 2, or 4). As for the quantization parameter (QP) of the H.264 intra-frames, it is chosen in such a way to permit a near-constant decoding quality in the output video sequence as in [9] and [10]. We notice that in the Foreman sequence (Figure 2), for the case of a fixed GOP size, the performance decreases as the GOP size increases. The best performance is thus obtained when all frames are intra-coded. This is due to the high motion in this sequence, which yields less accurate side information when the key frames are further apart. A similar effect has been noticed in [8] where key frames were encoded using an H.263+ video codec. However, when the GOP size is dynamically varied along the sequence, similar performance is obtained at low rates, compared to H.264 intra-coding, whereas a gain of 35 kbps is observed for a PSNR of 39.5 dB. A different behavior is observed with Grandmother, which is characterized by its relatively low motion, compared to the Foreman sequence. In the case of a fixed GOP, the best performance is obtained for a GOP size of 3, where the WZ codec outperforms H.264 intra coding at high rates. It can be seen, in Figure 3, that WZ coding with the proposed GOP size control algorithm outperforms both intra-coding and fixed-GOP WZ coding. For example, at a rate of 500 kbps, a gain of 0.8 dB is observed compared to H.264 intra-coding, and 0.3 dB compared to WZ coding with a GOP of size 3. Similarly, at 725 kbps, performance gains of 0.9 dB and 0.2 dB are obtained compared to H.264-Intra and best case WZ coding (GOP size of 3), respectively. In practical situations, the optimal GOP size cannot be known in advance, without effectively encoding and decoding the sequence with different GOP sizes. However, with our proposed GOP size control algorithm, the system is able to determine the optimal GOP size and, at the same time, improve the average R-D performance, since the GOP size is dynamically varied along the sequence, depending on the motion level in the video scene.

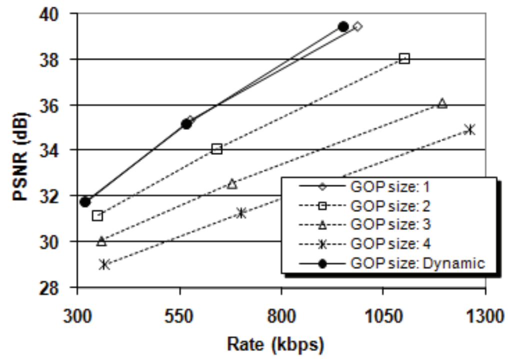

Fig. 2. Average R-D curves for the Foreman sequence using a WZ codec with fixed and variable GOP sizes.

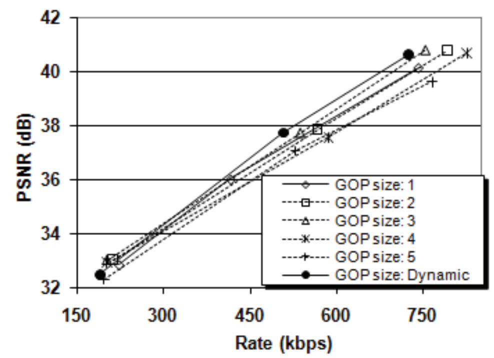

Fig. 3. Average R-D curves for the Grandmother sequence using a WZ codec with fixed and variable GOP sizes.

The variations of the ratio PSNR/R for fixed and adaptive GOP size coding are shown in Figure 4 for the Grandmother sequence, with M = 2. The plot shows only 80 frames for clarity of visualization. Points labeled with “I” indicate the beginning of a GOP (intra-frame) in the case of

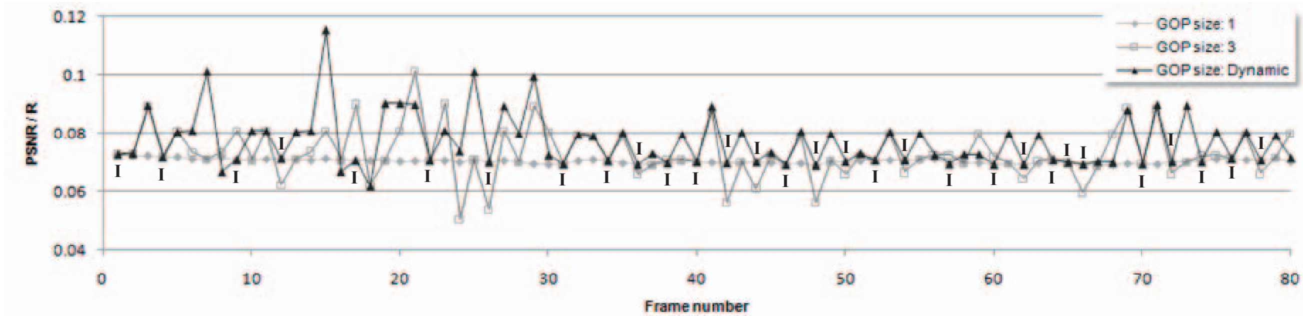

Fig. 4. Variations of the ratio PSNR/R for the Grandmother sequence, for M=2.

a dynamic GOP size. It can be clearly seen that, most of the time, the greatest ratio is obtained with the proposed algorithm. Only for a few frames, the best result is obtained with fixed GOP or intra-coding. This is expected since the proposed technique maximizes the ratio between the average PSNR and average bit-rate per GOP, whereas the plot is in terms of the frame number.

Figure 5 shows the percentage of GOP sizes obtained for each of the sequences using the proposed adaptive GOP size control algorithm, for different values of M. In the Foreman sequence, more than 80% of the GOPs are of size 1, which indicates that the system has switched to H.264 intra-coding mode most of the time. It is only in some rare regions that the system determines that a better performance can be obtained with WZ coding, rather than H.264 intra-coding, therefore slightly improving the average R-D performance, as shown in Figure 3. More GOP size variations can be observed with the Grandmother sequence: for M=1, not any GOP is of size 1, whereas L=5 for approximately 50% of the GOPs. For M=2 or 4, the most frequent GOP size is L=2 (40%).

4. CONLUSION

In this paper, we developed a novel algorithm for dynamically varying the GOP size in Wyner-Ziv video coding. A feedback channel is not needed for the decoding of WZ frames, and an automatic mode selection procedure allows the system to switch to H.264 intra-coding in regions where H.264 outperforms WZ coding, especially in high motion areas. A significant improvement in the overall performance was observed with the proposed technique, compared to intra-H.264 and fixed GOP WZ coding.

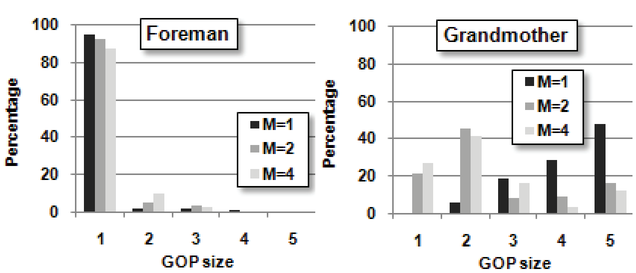

Fig. 5. Percentage of GOP sizes used in each sequence.

REFERENCES

[1] A. Aaron, R. Zhang and B.Girod, “Wyner-Ziv Coding of Motion Video”, 36th Asilomar Conference on Signals, Systems and Computers, pp. 240-244, November 2002.

[2] C. Yaacoub, J. Farah, B. Pesquet-Popescu, "Feedback Channel Suppression in Distributed Video Coding with Adaptive Rate Allocation and Quantization for Multiuser Applications", EURASIP Journal on Wireless Communications and Networking (WCN), 2008.

[3] X. Artigas and L. Torres, “Improved signal reconstruction and return channel suppression in Distributed Video Coding systems”, 47th International Symposium ELMAR-2005, Croatia, June 2005.

[4] M. Morbee, J. Prades-Nebot, A. Pizurica and W. Philips, “Rate Allocation Algorithm For Pixel-Domain Distributed Video Coding Without Feedback Channel”, 32nd International Conference on Acoustics, Speech and Signal Processing, Hawaii, USA, April 2007.

[5] D. Kubasov, K. Lajnef and C. Guillemot, “A hybrid encoder/decoder rate control for a Wyner-Ziv video codec with a feedback channel” , IEEE MultiMedia Signal Processing Workshop, Grece, October 2007.

[6] C. Brites and F. Pereira, “Encoder rate control for transform domain Wyner-Ziv video coding”, International Conference on Image Processing, USA, September 2007.

[7] Y. Tonomura, D. Shirai, T. Nakachi, and Tetsuro Fujii, "Optimal Bit Allocation for Wavelet-Based Distributed Video Coding", 8th IEEE International Symposium on Multimedia, USA, December 2006.

[8] A. Aaron, E. Setton and B. Girod, "Towards practical Wyner-Ziv coding of video", IEEE International Conference on Image Processing, Barcelona, Spain, Sept. 2003.

[9] J. Ascenso, C. Brites, F. Pereira, “Content Adaptive Wyner-Ziv Video Coding Driven by Motion Activity”, IEEE International Conference on Image Processing, Atlanta, USA, October 2006.

[10] C. Yaacoub, J. Farah, B. Pesquet-Popescu, “Optimal Rate Allocation in Multi-User Wyner-Ziv Video Coding Systems with Coded Key Frames”, 19th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, France, September 2008.

[11] D. Slepian and J.K. Wolf, “Noiseless Coding of Correlated Information Sources”, IEEE Transactions on Information Theory, Vol. IT-19, pp. 471-480, July 1973.

[12] D. Wyner and J. Ziv, “The Rate-Distortion Function for Source Coding with Side Information at the Decoder”, IEEE Transactions on Information Theory, Vol. IT-22, pp. 1-10, January 1976.

[13] ITU-T and ISO/IEC JTC1, “Advanced Video Coding for Generic Audiovisual Services,” ITU-T Recommendation H.264 – ISO/IEC 14496-10 AVC, 2003.

[14] C. Brites, J. Ascenso and F. Pereira, “Feedback Channel in Pixel Domain Wyner-Ziv Video Coding: Myths and Realities”, 14th European Signal Processing Conference, Italy, September 2006.