Multiple description video coding based on forward error correction within expanding windows

Chunyu Lin[1], Yao Zhao[1], Jimin Xiao [2,3], Tammam Tillo[3]
Institute of Information Science, Beijing Jiaotong University[1]
Beijing Key Laboratory of Advanced Information Science and Network, China[1]
Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, L69 3GJ, U.K.[2]
Department of Electrical and Electronic Engineering, Xian Jiaotong-Liverpool University, Suzhou, China[3]

***Abstract*—In this paper, an MDC scheme based on forward error correction(FEC) within expanding windows is proposed. Firstly, the video sequence will be coded into source packets with/without slice group enabled. Secondly, the appropriate FEC packets are inserted according to the packet loss rate. Since the previous frames in a GOP is generally more important than the following frames in the GOP, an expanding window is exploited so that the FEC packets for the current frame will also protect the previous frames in the window. After this, the source packets with the inserted FEC packets will be divided into two descriptions and transmitted into two independent channels. When some packets in one description are lost, FEC decoding will try to recover the lost packets. Through this scheme, the source packets can get appropriate protection while the compression efficiency will not be degraded too much. The experimental results show that the proposed scheme outperforms the compared schemes up to 3dB.**

***Keywords*-Video coding, Multiple description coding, forward error correction**

I. Introduction

When video bit-streams are transmitting through networks, even one packet loss will not only affect the decoding of the current frame but also degrade the other following frames' reconstructed quality due to the prediction and compensation structure. Multiple description coding (MDC) is an effective solution for such a problem. It can combat the packet loss without any retransmission, thus satisfying the real-time video communication. In MDC, one source is encoded into two or more bitstreams (descriptions), which are mutually refinable and can be decoded independently. When the network is reliable and all the descriptions are received, the best quality is obtained, which is usually referred to as the central performance. On the other hand, when only one description is received, it is still acceptable and referred to as side performance.

A lot of MDC schemes for robust video communication have been proposed, which can be classified as three main kinds. The first kind of schemes is based on polyphase subsampling, such as in spatial and temporal domain [1],[2]. This kind of scheme generally splits the signal into two or more sub-signals and encodes the sub-signals into descriptions. Due to the sub-sampling, the correlation between neighbor pixels or neighbor frames will be less than before, which results in the less compression efficiency. The second kind of schemes is based on scalar quantization that quantize the residual signal into two different descriptions [3]. Since the side description and central description have different reconstructed value, whether the prediction loop should be based on side value or central value will generate mismatch, which is a big problem for such kind of schemes. The last kind is based on the error resilient tools such as redundant picture(MDC-RP) [4], redundant slice (RS-MDC) [5] and redundant macroblock(MDC-RMB) [6]. For such kind of schemes, even though the redundant version could reduce the error propagation, too much redundancy are inserted. In addition, when the packet loss rate is large, the redundant part is similar to a duplicated version that has very large redundancy.

In this paper, an MDC scheme based on forward error correction (FEC) is proposed with appropriate inserted redundancy according to the network and the importance of each frame. To meet the requirement of real-time communication, systematic Reed Solomon (RS) code is employed so that the source packet will be kept untouched. In addition, RS code is applied across dependent frames with an expanding window because the error generally propagates from the current frame to the end of the GOP, which is the most important contribution of this paper.

II. The proposed scheme

Fig. 1 shows the coding process of our MDC scheme. Firstly, the video sequence is coded with H.264/AVC into two different slice groups frame by frame. For all the slices in each frame, appropriate parity packets are inserted using systematic RS code without differentiating the slice group number. After that, video packets in the same slice group together with half RS packets are saved into the two descriptions respectively. The details for each step is shown in the following.

A. Video packets dividing

In our scheme, each slice is encapsulated into one video packet. H.264 supports both slice and slice group, while the slice group further contains one or more slices. To divide the slices into two groups, we can use odd/even sub-sampling ways to separate the coded slices. Or we can employ the slice group. In H.264, there are 7 ways to form the slice group,

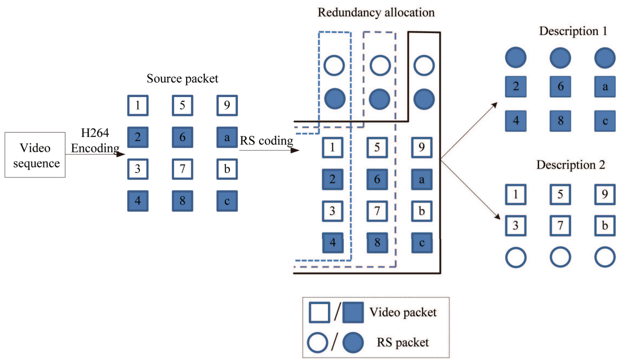

Fig. 1. The diagram of the proposed scheme.

which are interleave, dispersed, foreground with left-over, box-out, raster scan, wipe and explicit. With this technique, the lost packets can be concealed by exploiting the spatial redundancy of the images. On one hand, the correlation between different slice groups should be larger so that one of them can conceal the other when loss happens. On the other hand, the slice group will result in the less compression efficiency because the correlation in the slice group is less than before. For example, dispersed way scatters neighbor macroblocks that results in less spatial correlations in one slice group, while it increases the correlation between different slice groups. Hence, different ways should be selected according to the network status. Here, odd/even slice sub-sampling way is selected for low PLR($\leq 10\%$), while dispersed way is used for high PLR.

B. RS encoding and decoding within expanding windows

Systematic RS code is widely used for error and erasure correction. A RS code is specified as RS(n, k), where k represent the number of source packets, while $n-k$ denotes the number of parity packets and m is the number of bits in one symbol. In our case, if no less than k packets are received, RS decoding process can recover all the source packets. If less than k packets are received, the received source packets can still be used for video reconstruction. This is because the source packets are kept untouched during systematic RS encoding and each source packet contains one slice that can be decoded independently in H.264. The value of n and k can be any positive integer with the constraint as

$$\left\{ \begin{array}{c} k < n \\ n \leq 2^m - 1 \end{array} \right. \tag{1}$$

There are two things to be considered in RS(n, k) encoding. The first is that the efficiency of RS code is higher with larger value n [7]. However, there are not enough source packets in one video frames, especially for low resolution or low bit-rate case. If the source packets are taken form more than one frame, extra delay will be caused. The second is how many the number of parity packets should be assigned for each frame. Since the frames in the begin of a GOP will be referenced by the following frames, they should be protected more.

Hence, we propose to extend the RS encoding across dependent frames, as shown in Fig. 1. This expanding window concept is also used in the SVC and the layer aware coding scheme in which FEC is used to protect the dependent layer [8]. The source packets in the first frame will be protected with RS code normally. The source packets in the second frame, combined with the source packets in the first frame, will be protected together. All the source packets in the first three frames will be protected again at the instant of the third frame. Such a RS coding scheme protects the more important frame more. Take Fig. 1 as an example, assume three source packets are lost in the first frame, then the parity packets at this time cannot recover the lost packets because only two parity packets are inserted here. When decoding frame 2, the two parity packets in frame 1 will be combined with another two parity packets in frame 2 to do the RS decoding, which could recover the packet loss in frame 1. Even the corrupted frame 1 is already displayed for the users at this time, the recovered source packets could be used to update the reference so that the decoding of frame 2 will not be affected. Through this way, the frames in the begin of GOP get more protections with the expanding windows.

C. Initial redundancy allocation algorithm

The parity check packets can be seen as the inserted redundancy to protect the source packet. Allocating appropriate redundancy should depend on the network status and video importance. This section designs such an algorithm for redundancy allocation.

Assume the packet loss rate is p. After encoding one frame, the number of source packets can be known, supposed as k. According to the constraint (1), $n-k$ parity packets can be inserted which helps to resist the packet loss. To decide how many packets should be inserted, the residual packet loss rate is introduced here [7], [9].

$$\left\{ \begin{array}{c} p_r = \sum\limits_{m=n-k+1}^{n} \frac{m}{n} p(m) \\ p(m) = \binom{n}{m} p^m (1-p)^{n-m} \end{array} \right. \tag{2}$$

Where $p(m)$ denotes the probability that m packets are lost. The residual packet loss rate means that even using RS code, there is still unrecoverable probability denoted as p_r.

Therefore, enough parity packets should be inserted so that the residual packet loss rate for each frame is close to zero. In our case, it is more complicated to calculate the residual packet loss rate for each frame due to the expanding windows. Hence, we propose a simple recursive algorithm that processes from the first frame to the last frame in the expanding window. For the first frame, the accurate number of parity packets can be obtained to assure a very low residual packet loss rate. When encoding the second frame, the inserted parity packets are employed to protect both the source packets in the current frame and the source packet in the first frame. Since the residual packet loss in the first frame is very low, we can assume the source packets in this frame is recovered correctly. With this assumption, the residual packet loss rate for the second frame can be easily calculated, as well as for the other following frames in the windows. Hence, each frame will be assigned certain number of RS packets according to its residual packet loss rate.

D. Enhanced redundancy allocation algorithm

In the above redundancy allocation scheme, we assume that enough RS packets are provided to get a very low residual packet loss rate, which further assumes that the PLR is known for each frame. In practice, PLR can only be estimated statistically, which means that some frames could have more packet loss than its expected number, while the other frames could have less. In addition, there could be burst packet loss case. In fact, the packet loss can always be recovered at the end of the expanding window, if the actual PLR in the window is the same/smaller as/than its statistical value. As long as the window is long enough, the above condition can be met. However, when more packet loss occur in the first frame and it cannot be recovered until the end of the window, then the error in the first frame will propagate until the end of frame, which degrades the whole performance greatly. To solve this problem, some appropriate positions should be found to insert certain extra RS packets so that the total distortion is minimized. Suppose we know which position results the maximum distortion due to packet loss, then this position should deserve the first extra RS packet. However, calculating the distortion for each frame cannot be finished in real-time, especially due to the expanding windows introduced. Hence, we introduce some approximation here.

Assume the packet loss happens randomly and each frame consists of the same number of packets, that is, any frame could have a more packet loss than its expected number, where a is an integer. This assumption is generally true if I frame is not considered and only P frame is used. Actually, the number of slices in each P frames does not fluctuate too much. The distortion for each frame is composed of two parts, while the first part is resulted from the current frame due to the packet loss and the second part is caused by the error propagation from current frame to the following frames. To estimate the amount of propagated distortion, a powder transfer function $f(l)$ is employed [5], l is the distance from the error occurred frame. If $d(l)$ is used to represent the distortion due to the packet loss at current frame l, then the propagated distortion $d_p(l)$ can be defined as

$$d_p(l) = \sum_{i=1}^{w-l} d(l) * f(i) \tag{3}$$

Where w is the length of the expanding window. This formula means that the distortion $d(l)$ will propagate from current frame l until the end of window w with its corresponding weight $f(i)$. For simplicity, the function $f(l)$ is employed as

$$f(i) = e^{-0.4i} \tag{4}$$

Since $f(0) = 1$, the total distortion resulted from current packet loss is

$$d_t(l) = d(l) + d_p(l) = \sum_{i=0}^{w-l} d(l) * f(i) \tag{5}$$

However, the calculation of $d_t(l)$ for each frame can only be obtained after encoding the corresponding frame, which means we cannot get it at the instant of encoding the first frame. Here we assume each packet loss results in the same amount of distortion for its corresponding frame, defined as d_c. Then the total distortion for each frame can be obtained before its encoding. Consequently, where the first extra RS packet should be inserted is determined so that the maximum distortion can be avoided. After that, the total distortion for each frame should be recalculated to find another position that has the maximum distortion, which deserves the second RS packet. The whole algorithm is shown as

Algorithm 1: Extra RS packets allocation algorithm

Given w the window length, l the current frame
Given $d(l)$ the distortion caused by packet loss,
Given $d_p(l)$ the distortion caused by error propagation
while (not the last extra RS packet)
  1 for $i = 1 : w$ **do**
      calculate $d_t(i) = d(i) + d_p(i)$
    end for
  2 find the position that has maximum d_t,
    assign the RS packet there,
    go to **1**
end while

Notice $d(l)$ is used here for a general form. To make it real-time, d_c should be employed.

III. Experimental results and analysis

In this section, the proposed scheme are compared with the state-of-the-art MDC schemes. The GOP structure is IPPP, the frame rate is 30 fps and the GOP length is 30. The number of reference frame is 5 and the slice size is fixed as 400 bytes. Finally, the length of the expanding window is selected as the GOP length. Two CIF video sequences, *Foreman* and *Coastguard* are used in the simulation. MDC scheme with Redundancy allocation at Macro Block level(MDC-RMB) [6] is one of classical schemes that outperforms that of RS-MDC[5] and MDC-RP[4], which is selected as an anchor. To provide a fair comparison, the first 90 frames are selected when compared with MDC-RMB and the Quantization Parameter (QP) ranges in [22:2:38]. The PLR is selected in the range of {0.05, 0.10, 0.20}, in which 20% is taken as the higher PLR and slice group with dispersed way is used. From Fig. 2 and Fig. 3, it can be seen that the proposed scheme outperforms MDC-RMB significantly, which is mainly due to the expanding window and the efficient redundancy allocation. The expanding window assures the error can be corrected with high probability, while the redundancy allocation scheme provides appropriate protection with less degraded compression efficiency. In Fig. 4, another MDC scheme employing Joint Temporal and Spatial Error Concealment(MDC-JTSEC)[1] is

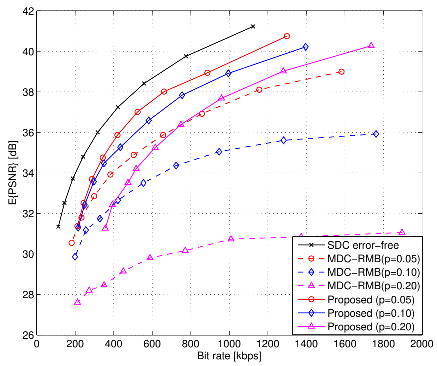

Fig. 2. Expected PSNR results compared with MDC-RMB(Foreman CIF, the first 90 frames).

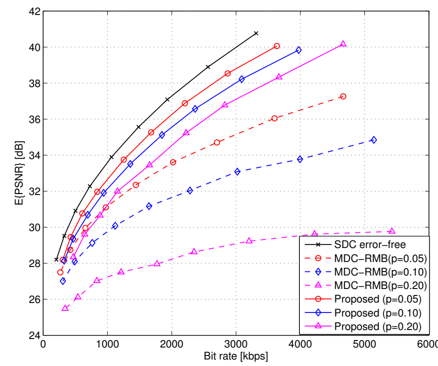

Fig. 3. Expected PSNR results compared with MDC-RMB(Coastguard CIF, the first 90 frames).

used for comparison because it outperforms the classical schemes, such as the hybrid MDC [2]. The PLR is selected in the range of $\{0.01, 0.05, 0.10\}$. Notice that every packet consists of one-fourth information of one original frame in MDC-JTSEC scheme(fixed Macroblock), which is different from the fixed size of the proposed scheme. Hence, MDC-JTSEC has advantage when the PLR is lower. It can be seen that the proposed scheme is better than that of MDC-JTSEC at all packet loss conditions. Above 3dB gain at PLR=0.01 shows the great advantage of the proposed scheme. All the results show that the gain is larger with the increasing of PLR, which shows the great error resiliency performance of the proposed scheme. It can also be noted that the gain is smaller at lower bit-rate because RS coding efficiency is lower with fewer source packets.

The proposed scheme is promising for its good performance, however, the computation is quite high. To solve this problem, the length of the window can be shortened. In fact, with 3 frames in one window, the proposed scheme is still very efficient, about 1dB lower than that of GOP length. The whole results demonstrate the good performance of the proposed schemes.

IV. Conclusion

In this paper, a multiple description video coding scheme based on forward error correction within expanding windows is proposed. With appropriate FEC packets inserted and the correlated protection in the expanding windows, the proposed scheme outperforms the compared state-of-the-art MDC schemes significantly. It should be noted that the RS coding and decoding is complex. Hence, the further work will consider the speeding up of the whole scheme.

Acknowledgment

This work was supported by Major Program of National Natural Science Foundation of China (61210006), PCSIRT(IRT201206), NSFC(61202240), the Fundamental Research Funds for the Central Universities(2013JBM018, 2012RC043).

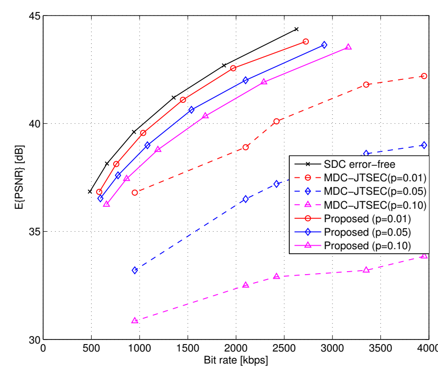

Fig. 4. Expected PSNR results compared with MDC-JTSEC(Foreman CIF, the whole sequence).

References

[1] W.-J. Tsai and J.-Y. Chen, "Joint Temporal and Spatial Error Concealment for Multiple Description Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 12, pp. 1822 –1833, Dec. 2010.

[2] C.-W. Hsiao and W.-J. Tsai, "Hybrid Multiple Description Coding Based on H.264," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 1, pp. 76 –87, Jan. 2010.

[3] O. Campana, R. Contiero, and G. Mian, "An H.264/AVC Video Coder Based on a Multiple Description Scalar Quantizer," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 2, pp. 268 –272, Feb. 2008.

[4] I. Radulovic, P. Frossard, Y.-K. Wang, M. Hannuksela, and A. Hallapuro, "Multiple Description Video Coding with H.264/AVC Redundant Pictures," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 20, no. 1, pp. 144–148, Jan. 2010.

[5] T. Tillo, M. Grangetto, and M. Olmo, "Redundant Slice Optimal Allocation for H.264 Multiple Description Coding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 18, no. 1, pp. 59–70, Jan. 2008.

[6] C. Lin, T. Tillo, Y. Zhao, and B. Jeon, "Multiple Description Coding for H.264/AVC With Redundancy Allocation at Macro Block Level," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 5, pp. 589 –600, May 2011.

[7] J. Xiao, T. Tillo, C. Lin, and Y. Zhao, “Dynamic Sub-GOP Forward Error Correction Code for Real-Time Video Applications,” *IEEE Transactions on Multimedia*, vol. 14, no. 4, pp. 1298 –1308, aug. 2012.
[8] C. Hellge, D. Gomez-Barquero, T. Schierl, and T. Wiegand, “Layer-Aware Forward Error Correction for Mobile Broadcast of Layered Media,” *IEEE Transactions on Multimedia*, vol. 13, no. 3, pp. 551 –562, June 2011.
[9] L. Kondrad, I. Bouazizi, and M. Gabbouj, “Media aware FEC for Scalable Video Coding transmission,” in *IEEE Symposium on Computers and Communications*, July 2009, pp. 7 –12.