Multiple Description Coding of Flash Video based on Adaptive Allocation of DCT Coefficients

Akinori Ito, Takuya Kuraishi, Masashi Ito and Shozo Makino
Graduate School of Engineering, Tohoku University
6-6-05 Aramaki aza Aoba, Sendai, 980-8579 Japan
{aito,kura,itojin,makino}@makino.ecei.tohoku.ac.jp

***Abstract*—In this paper, we propose a method for multiple description coding (MDC) of Flash Video stream (FLV). Our target codec of FLV is Sorenson H.263. Conventional MDC methods had disadvantages that they required large redundancy. We proposed a method that considers "patterns" of a macroblock, and it changes how to treat DCT coefficients of a macroblock according to the pattern. As an experimental result, we could reduce redundancy of the encoded stream while keeping the video quality.**

I. Introduction

Video streaming services such as YouTube become more and more popular. Most of these services are based on Flash Video, which can be viewed using a web browser because the flash viewer is provided as a plug-in of major web browsers. Current implementation of Flash Video is based on TCP, which automatically retransmits the lost packets. Although communication using the TCP is reliable, there are two drawbacks when using it as a communication protocol of video streaming. One is that the TCP is not suitable for real-time communication, because we cannot control or predict interval of packets because of packet re-transmissions. The other one is that the TCP is not suitable for broadcasting. As the TCP is a protocol for one-to-one transmission, there must be as many connections as the number of clients when broadcasting video. This causes severe server load when broadcasting video to thousands of clients simultaneously.

One solution of these problems is to use the RTP as a transmission protocol instead of the TCP. As the RTP does not re-submit lost packet, interval of received packets roughly coincides that of the sent packets[1]. Moreover, we can use multicast for sending the video stream to many clients, where the server send just one stream for delivering video.

On using the RTP as a transmission protocol, we have to consider how to deal with packet losses. As the UDP or RTP does not re-transmit the lost packet, the application must conceal the packet losses so that quality of the signal does not severely degrade. Various methods for concealing packet losses have been proposed so far [2]. Multiple description coding is one of the methods for packet loss concealment, which enables high-quality packet loss concealment with relatively small amount of side information [3].

In this paper, we describe a new multiple description coding method for Flash Video stream. Our target is an FLV stream encoded with Sorenson H.263 codec, and the bitstream is split into two descriptions. We also describe the experimental result for comparing the proposed method with the MD-split method[4].

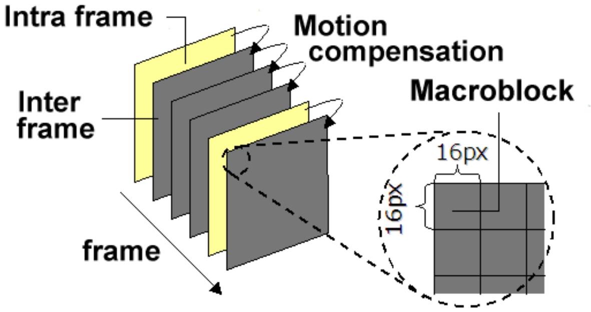

Fig. 1. Structure of Sorenson H.263

II. FLV and Sorenson H.263

First, we explain our target, FLV and Sorenson H.263. FLV [5] is a container used for Flash Video stream, which can include audio and video frames. One FLV stream has one FLV header and FLV tags, each of which contains either a video frame or an audio frame. The screenvideo, Sorenson H.263, On2VP6 and H.264 are available as codecs for video stream for FLV. In this work, we targeted the Sorenson H.263 codec, which is the most popular one among those codecs for Flash Video. Sorenson H.263 is a subset of ITU-T H.263. Figure 1 shows the basic scheme of Sorenson H.263 codec. A video frame is either an intra frame or an inter frame, where an intra frame is encoded independently from the past frames, while an inter frame is encoded so that only differences from the past frame are encoded.

One frame is divided into macroblocks, each of which has 16 × 16 pixels. Pixels in a macroblock are analyzed using the discrete cosine transform (DCT), as well as motion compensation using motion vectors (MV). The analyzed data in a macroblock are quantized and compressed using an entropy coding, and finally appended to a header for composing a packet. Thus, a bit pattern of one macroblock has a header, MV and DCT information. As explained later, in some cases the MV and/or DCT are omitted.

III. Related works

Several methods have been proposed for multiple description coding for video stream. These methods are divided into two groups; one group is for methods that are independent from codec, and the other group is for codec-dependent methods.

As a codec-independent method, Apostolopoulos proposed a method that split a video stream into two descriptions temporally, which encoded the even-numbered frames and odd-numbered frames independently [6]. Vitali proposed a method that splits a video stream spatially, where four neighboring pixels were transmitted into four independent channels [7]. An advantage of these methods is that they do not depend on the codec. However, a disadvantage of the codec-independent methods is that the encoding becomes less efficient when dividing the original video stream into descriptions, because we have to encode each description without using dependency between descriptions.

As a method depending on the ITU-T H.263 codec, Reibman et al. developed the MD-split method that splits H.263 bitstream into two descriptions[4]. In their method, the header and MV are copied to both descriptions. As for DCT, the coefficients with large values are copied to the both descriptions. When a DCT coefficient is smaller than the threshold, it is copied to only one of two descriptions by turns, and the value "one" is copied to the other description instead of the original value. In general, a codec-dependent MDC is more efficient than a codec-independent one. The MD-split method keeps better quality even when packets are lost than a method using correlating transform [8]. However, it still requires more than 1.6 times larger bitrate than the original bitrate.

IV. The pattern-adaptive MD coding

A. Macroblock pattern of Sorenson H.263

One reason why the MD-split method requires much redundancy is that their method does not take "real" importance of a DCT coefficient into account. They regard the coefficients with large values as important ones, but it is not always true.

The importance of DCT differs from macroblock to macroblock. To explain this, we first explain the "patterns" of macroblocks in Sorenson H.263.

As explained before, a bit pattern of a macroblock contains a header, an MV and DCT coefficients. The header is composed from three fields: COD (coded macroblock indication), MCBPC (macroblock type & coded block pattern for chrominance) and CBPY (coded block pattern for luminance). COD is a one-bit field, and MCBPC and CBPY are coded by a variable-length code. There are 11 possibilities of combinations of these fields in the ITU-T H.263. As the Sorenson H.263 is a subset of ITU-T H.263, we have only four "patterns" of combinations of these fields in a header as well as MV and DCT. Table I shows the four patterns.

TABLE I
Presence of data for the four "patterns"

Pattern	COD	MCBPC	CBPY	MV	DCT
1	0	yes	yes	yes	yes
2	1	no	no	no	no
3	0	yes	yes	no	yes
4	0	yes	yes	yes	no

Pattern 1: This pattern in the most common one. A bit pattern has all of a header, an MV and a DCT. In this pattern, differences of pixel values from the previous frame are coded in the DCT block.

Pattern 2: This pattern only has the COD bit. When this pattern is used, the decoded macroblock is identical with the macroblock at the same position in the previous frame.

Pattern 3: This pattern does not have an MV. When this pattern is used, the input macroblock is coded into DCT without referring the previous frame.

Pattern 4: This pattern does not have a DCT. When this pattern is used, the decoded macroblock is generated only by applying the motion compensation from the previous frame.

Among these four patterns, only pattern 1 and 3 have DCT. It should be noted that the DCT in the pattern 3 is not generated from differences of the pixel values but generated from the pixel values themselves. This means that the macroblock cannot be constructed at all when the DCT of pattern 1 is lost. Conversely, when a macroblock is coded in pattern 1, the macroblock can be recovered with a little degradation even when the DCT is lost because the macroblock can be estimated using the previous frame and MV.

B. Pattern-adaptive MD coding

Considering the above difference between pattern 1 and 3, we developed an MD coding method. Our method splits the input video stream into two descriptions as follows:

1) A header in a macroblock is copied to the both descriptions.
2) If the macroblock has an MV, it is copied to the both descriptions.
3) If the macroblock has a DCT, it is processed as follows:
   a) If the macroblock is pattern 3, all coefficients of the DCT are copied to the both descriptions.
   b) If the macroblock is pattern 1, all coefficients of the DCT are copied to one of two descriptions in turn. The DCT is also copied to the other description with probability p, or no DCT is copied to the other description.

Here, the probability p is used to control the tradeoff between redundancy and quality. If $p = 0$, the DCTs of macroblocks of pattern 1 are copied to only one of two descriptions; if $p = 1$, the two descriptions are just exact copies of the original video stream.

When one of two descriptions is lost, the decoder decodes the macroblock using information contained in the remaining description. If the pattern of the macroblock is other than 1, all information needed for decoding is contained in the both description, and thus the original macroblock can be recovered from only one description. Conversely, when the pattern of the macroblock is 1, we lose the DCT coefficients with probability $(1-p)/2$. If the DCT is lost, we estimate the macroblock as follows. Let $b_k(t)$ be the k-th macroblock of the t-th frame. Suppose one description of $b_k(t)$ is lost and we lose DCT information.

1) If both $b_k(t-1)$ and $b_k(t+1)$ are not lost and their pattern is 1, then we estimate the DCT of $b_k(t)$ by averaging those of $b_k(t-1)$ and $b_k(t+1)$.
2) Otherwise, we regard $b_k(t)$ as pattern 4, where only MV is used for restoring the macroblock.

V. Experiment

A. Experimental conditions

An evaluation experiment was carried out. We used three standard video clips, "foreman," "football" and "mobile" [9] as test materials. These video clips were converted to 352×288 YUV420 format at 30 fps, and then encoded to Sorenson H.263 format.

The quality of the encoded video under a certain packet loss condition was measured using distortion, which is a difference of average PSNR of the video without packet loss and that with packet losses. PSNR was calculated as follows:

$$PSNR\ \text{(dB)} = 10\log_{10}\frac{255^2}{\frac{1}{3N}\sum_n \Delta YUV_n^2} \tag{1}$$

$$\Delta YUV_n^2 = (Y_n - Y_n')^2 + (U_n - U_n')^2 + (V_n - V_n')^2 \tag{2}$$

Here, $0 \leq n < N$ where N is the number of pixels $(352 \times 288 = 101376)$. Y_n, U_n and V_n are the n-th pixel values of Y, U and V plane of the original video, and Y_n', U_n' and V_n' are those of the degraded video. Note that we have only one pixel value for neighboring four pixels in U and V planes; on calculating PSNR, we used the same values as pixel values of the four pixels corresponding one value in U and V planes. Then the distortion is calculated as

$$D = PSNR_O - PSNR_L \tag{3}$$

where $PSNR_O$ is the average PSNR of the encoded video without packet losses and $PSNR_L$ is that with packet losses. D becomes zero when no packet loss occurs, and it becomes larger when the degradation is severe.

The other measure of the method is redundancy, which is calculated as:

$$R = \frac{B_M}{B_O} - 1 \tag{4}$$

where B_O is the average bitrate of the video stream encoded in Sorenson H.263 codec and B_M is that of all video streams generated by the MD coding.

B. Redundancy and distortion

Figure 2,3 and 4 show Redundancy-Rate-Distortion (RRD) curves of the three video clips when one of two descriptions is completely lost (i.e. 50% packet loss). When redundancy is more than 50%, distortion by the proposed method (Adaptive MDC) and that by the MD-split method are almost same. However, the proposed method can reduce the redundancy to 10%–35% without severe degradation.

C. Packet loss rate and distortion

Next, we investigated influence of packet loss rate on the quality of restored video stream. In this experiment, we exploited the Gilbert loss model and the average length of packet losses was set to 3. We assumed that at least one description corresponding one macroblock could be received. In the proposed method, the probability p was set to 0.5; in the MD-split method, threshold for DCT allocation was set to 1000. Figure 5, 6 and 7 shows the average distortion with respect to packet loss rate. This result proves that the proposed method shows better quality regardless of packet loss rate.

VI. Conclusion

In this paper, we proposed a method for improving quality of Flash Video (Sorenson H.263) when transmitted through channels with packet losses. The proposed method allocates DCT coefficients of the original video stream into two descriptions considering pattern of macroblocks. When a macroblock contains DCT of differences of pixel values, the DCT block is omited according to a probability for reducing bitrate. From the experimental result, it was shown that the proposed method outperformed the conventional method for any packet loss rate.

References

[1] Colin Perkins, *RTP: Audio and Video for the Internet*, Addison-Wesley, 2008.

[2] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication : A review," *Proc. IEEE*, vol. 86, pp. 974–997, 1998.

[3] V. K. Goyal, "Multiple description coding: Compression meets the network," in *IEEE Signal Processing Magazine*, 2001, pp. 74–93.

[4] A. Reibman, H. Jafarkhani, Yao Wang, and M. Orchard, "Multiple description video using rate-distortion splitting," in *Proc. Int. Conf. on Image Proc.*, 2001, vol. 1, pp. 978 – 981.

[5] Adobe Systems Inc., "Macromedia flash (swf) and flash video (flv) file format specification version 8," 2005.

[6] J. G. Apostolopoulos, "Reliable video communication over lossy packet networks using multiple state encoding and path diversity," in *Visual Communications and Image Processing*, 2001.

[7] Andrea Vitali, "Multiple description coding - a new technology for video streaming over the internet," EBU Technical Review, Oct. 2007.

[8] A. Reibman, H. Jafarkhani, Y. Wang, M. T. Orchird and R. Puri, "Multiple description coding for video using motion compensated prediction," IEEE Trans. Circuits and Systems for Video Tech., vol. 12, no. 3, pp. 193–204, 2002.

[9] The Video Quality Experts Group, http://www.its.bldrdoc.gov/vqeg/.

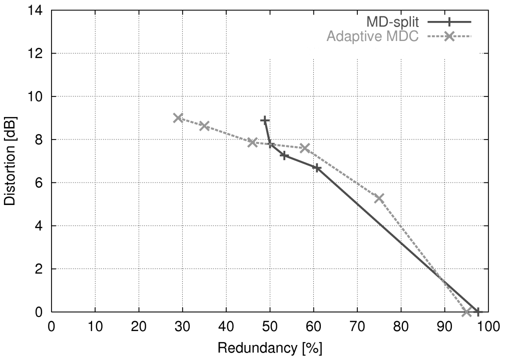

Fig. 2. RRD curves for three video clips (football)

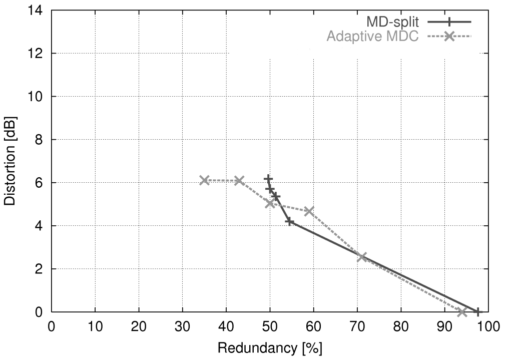

Fig. 3. RRD curves for three video clips (foreman)

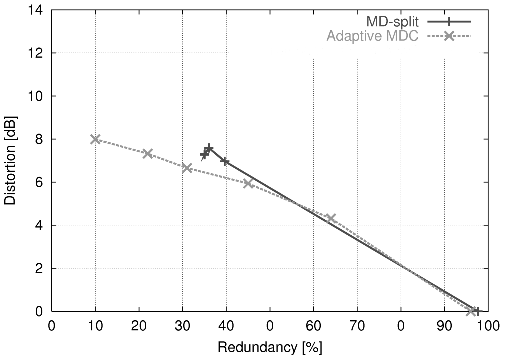

Fig. 4. RRD curves for three video clips (mobile)

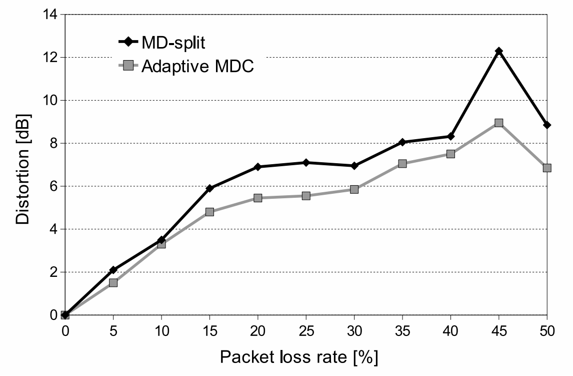

Fig. 5. Distortion with respect to packet loss rate (football)

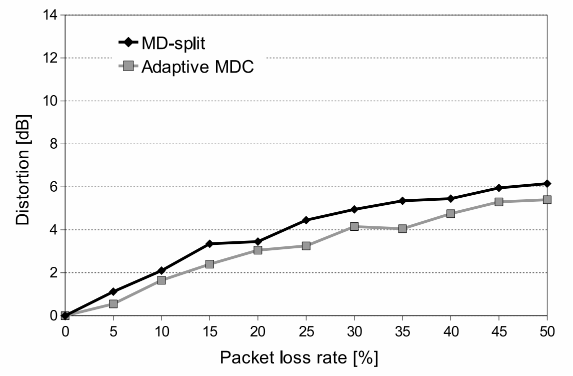

Fig. 6. Distortion with respect to packet loss rate (foreman)

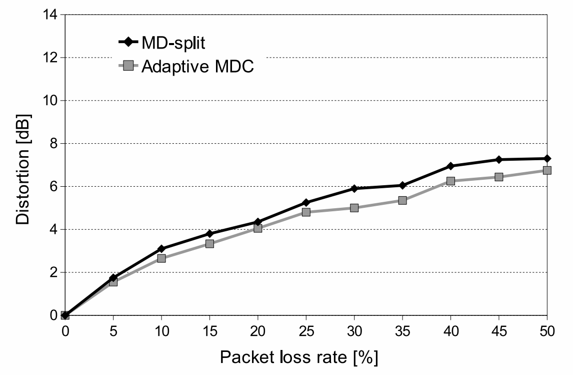

Fig. 7. Distortion with respect to packet loss rate (mobile)