A SIMPLIFIED DUAL-BITSTREAM MPEG VIDEO STREAMING SYSTEM WITH VCR FUNCTIONALITIES

Tak-Piu Ip, Yui-Lam Chan, Chang-Hong Fu and Wan-Chi Siu

Centre for Multimedia Signal Processing
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

ABSTRACT

Nowadays, video playback devices have only limited fast-forward/backward playback and even they cannot provide backward playback. The limitation is due to the motion compensated prediction technique adopted in the MPEG standards. One possible way to support browsing functionalities is to store an additional reverse-encoded bitstream into the server. However, this additional bitstream increases the storage requirement of the video server significantly. In this paper, we exploit the redundancy inherent between the forward and reverse-encoded bitstreams in order to achieve a substantial reduction on the size of the reverse-encoded bitstream. The server accesses and manipulates various macroblocks from the forward and reverse-encoded bistreams to facilitate various browsing operations. Experimental results show that, as compared to the conventional dual-bitstream scheme, the new scheme significantly reduces the storage requirement due to the additional reverse-encoded bitstream.

Index Terms— Digital video cassette recording (VCR), digital video browsing, MPEG video, streaming video, dual-bitstream scheme.

1. INTRODUCTION

Current video compression standards [1-3] such as MPEG are basically developed for the purposes of efficient video storage and transmission, not browsing. Nowadays, video playback devices offer relatively few controls for video browsing. Its limitation is due to the motion compensated prediction technique [4-5] adopted in the MPEG standards. This technique is mainly designed for normal forward playback. The MPEG video data is then not invariant to changes in frame order and it makes some video cassette recording (VCR) operations difficult.

Recently, a backward-play transcoder [6-7] has been described to convert a sequence with I–P frames into another I–P bitstream with reverse frame order. It still requires much computation. We have also developed some compressed-based techniques [8] in the server to provide backward playback over a network by reducing the required network bandwidth and decoder effort of VCR requests. Lin et al. [9] recently proposed to store the forward-encoded bitstream (FB) and the reverse-encoded bitstream (RB) in the server. The idea behind is to switch frames between the FB and the RB in order to minimize the transmitted frames over a network for any speed-up factors. This dual-bitstream technique can alleviate the decoder complexity while maintaining the low network bandwidth requirement on VCR operations. However, an extra storage is required for the RB. In this paper, a simplified RB is proposed to alleviate the storage requirement of the RB.

The organization of this paper is as follows. Section 2 of this paper reviews the dual-bitstream scheme. The proposed simplified RB used in the dual-bitstream video streaming system is then described in Section 3. Experimental results are presented in Section 4. Finally, some concluding remarks are provided in Section 5.

2. DUAL-BITSTREAM MPEG VIDEO STREAMING SYSTEM

In [9], a dual-bitstream technique was proposed to cope with the problem in different VCR trick modes of the MPEG video streaming system. This dual-bitstream technique adds a RB in the server in addition to the traditional FB. The generation of the RB can simply be done by encoding the video in reverse order. Fig. 1 shows an illustrative example of the FB and RB in which the video is coded in I-/P-frames with a GOP size of 14 frames. Note that the I-frames in the RB are interleaved between I-frames in the FB. Upon the server receives the VCR command or the requested frame number from the client, the server employs the frame-selection strategy to determine which frames in either the FB or RB should be transmitted to the client by minimizing the number of frames to be sent over the network. To illustrate the scheme, let us use the structure of dual bitstreams in Fig. 1. Assume that

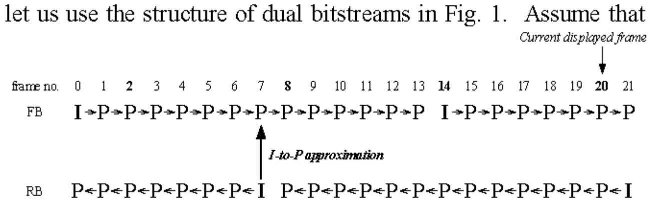

Fig. 1. The dual-bitstream scheme: Fast-backward operation with a speed-up ratio of 6.

the previous mode was in the normal forward play at frame 20 and the requested mode is fast-backward playback with a speed-up factor of 6. This operation requires to display frames 14, 8, etc. If the requested frame is an I-frame in one of two bitstreams, the frame can be decoded by itself. Thus, in the above example, frame 14 will be decoded from the FB directly since it is an I-frame. Then, the next frame to be decoded is frame 8. Since the requested frame is a P-frame in both bitstreams, the current displayed frame, or the nearest I-frame either in the FB or the RB is firstly selected to initiate the decoding of the requested frames. In this example, frame 8 will be decoded from frame 7 of the RB (an I-frame) since the nearest I-frame of the RB (frame 7) is the closest reference to frame 8. Note that frame 7 of the RB (an I-frame) is used as an approximation of frame 7 of the FB (a P-frame) to reconstruct frame 8 of the FB, as depicted in Fig. 1. This I-to-P approximation would cause the problem of reference mismatch in the reconstructed frame. It further causes drift when the approximated frames are used as the reference frames to predict the following P-frames and will last until the next I-frame. However, the subjective degradation is not significant visually due to the fast changes of the video content displayed [10].

3. SIMPLIFIED RB (SRB) IN THE DUAL-BITSTREAM SCHEME

As mentioned above, the dual-bitstream scheme can provide an effective way to support VCR operations for MPEG video. However, it requires extra storage for the RB. In this paper, a technique for reducing the storage requirement of the RB is suggested. The proposed technique attempts to exploit redundancy in some MBs found between the two bitstreams.

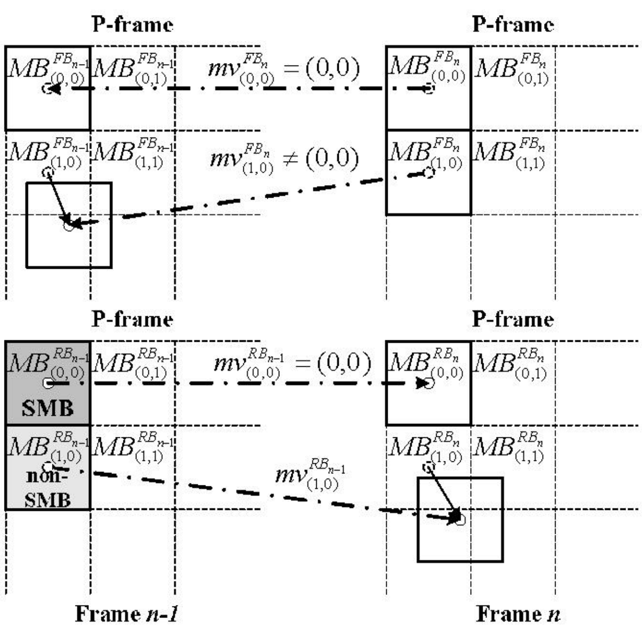

Fig. 2. Macroblock views of the proposed dual-bitstream and the definition of the SMB and non-SMB.

The situation in MB level of the dual-bitstreams is depicted in Fig. 2. In the server, $MB_{(k,l)}^{FB_n}$ and $MB_{(k,l)}^{RB_n}$ represent the MBs at the k^{th} row and l^{th} column of frame n in the FB and RB respectively.

To reduce the temporal redundancy in coding video sources, block motion-compensated prediction is used in which the previously transmitted and decoded frame serves as the prediction for the current frame. The difference between the prediction and the actual current frame is then the prediction error. The prediction errors in the FB, $e_{(k,l)}^{FB_n}$, and the RB, $e_{(k,l)}^{RB_{n-1}}$, are given by

$$e_{(k,l)}^{FB_n} = MB_{(k,l)}^{FB_n} - MCMB^{FB_{n-1}}(mv_{(k,l)}^{FB_n}) \tag{1}$$

and

$$e_{(k,l)}^{RB_{n-1}} = MB_{(k,l)}^{RB_{n-1}} - MCMB^{RB_n}(mv_{(k,l)}^{RB_{n-1}}) \tag{2}$$

where $MCMB^{FB_{n-1}}(mv_{(k,l)}^{FB_n})$ stands for the motion-compensated MB of $MB_{(k,l)}^{FB_n}$ which is translated by the motion vector $mv_{(k,l)}^{FB_n}$ in frame *n-1* of the FB and $MCMB^{RB_n}(mv_{(k,l)}^{RB_{n-1}})$ represents the motion-compensated MB of $MB_{(k,l)}^{RB_{n-1}}$ with the displacement of the motion vector $mv_{(k,l)}^{RB_{n-1}}$ in frame n of the RB. Note that, in contrast to the FB, frame *n-1* is predicted from frame n in the RB since this bitstream is generated by encoding the video frames in reverse order. All these prediction errors are transformed in the discrete cosine transform (DCT) domain. The DCT coefficients are then quantized, variable-length encoded and stored in the server.

In a video sequence, frames at the same time instant in the FB and RB are perceptually similar to each other. They actually represent the same contents and have similar color, texture, and objects, but the only difference is the coding directions, as described in (1) and (2). This means if the RB is encoded completely as a separate bitstream from the FB, a considerable amount of redundancy exists. To generate the RB in the proposed dual-bitstream scheme, the strategy is to reuse the MB data as much in the FB as possible. For this purpose, a special measure is taken to encode some MBs in the RB which can utilize the MB data in the FB. In the proposed technique, all coefficients of these MBs are not necessary to be encoded and they are defined as skipped MBs (SMBs) in the RB. $MB_{(k,l)}^{RB_{n-1}}$ is defined as a SMB if the MB in frame n of the FB, $MB_{(k,l)}^{FB_n}$, is coded without motion compensation (non-MC MB). Otherwise, it is defined as a non-SMB. For illustration, we use the example in Fig. 2 again to give a clear account of the definition of the SMB. In this figure, since the motion vector of $MB_{(0,0)}^{FB_n}$, $mv_{(0,0)}^{FB_n}$, is zero, it means that $MB_{(0,0)}^{FB_n}$ is a non-MC MB and $MB_{(0,0)}^{RB_{n-1}}$ is classified as a SMB. On the other hand, since $MB_{(1,0)}^{FB_n}$ is coded with motion compensation (MC MB), $MB_{(1,0)}^{RB_{n-1}}$ is classified as a non-SMB.

When $MB_{(k,l)}^{RB_{n-1}}$ in the RB is found to be a SMB, its corresponding MB in frame n of the FB, $MB_{(k,l)}^{FB_n}$, is coded without motion compensation. It means that the spatial position of $MB_{(k,l)}^{FB_{n-1}}$ in the FB is the same as that of $MB_{(k,l)}^{FB_n}$. Hence, for this specific case, $MCMB^{FB_{n-1}}(mv_{(k,l)}^{FB_n})$ is equal to $MB_{(k,l)}^{FB_{n-1}}$, and (1) can be rewritten as

$$e_{(k,l)}^{FB_n} = MB_{(k,l)}^{FB_n} - MB_{(k,l)}^{FB_{n-1}} \tag{3}$$

In order to reuse the MB data as much in the FB as possible during encoding $MB_{(k,l)}^{RB_{n-1}}$ of the RB, its motion vector is enforced to zero as well, i.e., $mv_{(k,l)}^{RB_{n-1}} = 0$. Such arrangement is to ensure $MCMB^{RB_n}(mv_{(k,l)}^{RB_{n-1}})$ is equal to $MB_{(k,l)}^{RB_n}$ such that (2) becomes

$$e_{(k,l)}^{RB_{n-1}} = MB_{(k,l)}^{RB_{n-1}} - MB_{(k,l)}^{RB_n} \quad (4)$$

As mentioned before, frame n of the FB and RB share the same video content. Because of this, pixels of $MB_{(k,l)}^{FB_n}$ and $MB_{(k,l)}^{RB_n}$ are similar and it is reasonable to approximate $MB_{(k,l)}^{RB_n}$ by $MB_{(k,l)}^{FB_n}$ during various VCR operations. That is,

$$MB_{(k,l)}^{RB_n} \approx MB_{(k,l)}^{FB_n} \quad (5)$$

Similarly,

$$MB_{(k,l)}^{RB_{n-1}} \approx MB_{(k,l)}^{FB_{n-1}} \quad (6)$$

By putting (5) and (6) into (4), it can be rewritten as

$$e_{(k,l)}^{RB_{n-1}} = MB_{(k,l)}^{FB_{n-1}} - MB_{(k,l)}^{FB_n} \quad (7)$$

From (3) and (7), we get

$$e_{(k,l)}^{RB_{n-1}} = -e_{(k,l)}^{FB_n} \quad (8)$$

By applying the DCT to (8) and considering that the DCT is an odd transform, we can obtain $e_{(k,l)}^{RB_{n-1}}$ in the DCT domain as indicated below,

$$DCT(e_{(k,l)}^{RB_{n-1}}) = -DCT(e_{(k,l)}^{FB_n}) \quad (9)$$

Then the quantized DCT coefficients of $e_{(k,l)}^{RB_{n-1}}$ are computed as

$$Q[DCT(e_{(k,l)}^{RB_{n-1}})] = -Q[DCT(e_{(k,l)}^{FB_n})] \quad (10)$$

$Q[DCT(e_{(k,l)}^{RB_{n-1}})]$ is the quantized DCT coefficients to be encoded in the RB. However, $Q[DCT(e_{(k,l)}^{FB_n})]$ is already available in the FB. From (10), $Q[DCT(e_{(k,l)}^{RB_{n-1}})]$ can be extracted directly from the FB by simply inverting the signs of all quantized DCT coefficients in $Q[DCT(e_{(k,l)}^{FB_n})]$. Therefore, the server can store the simplified RB (SRB) instead of the RB. SRB is the one that video data about the SMBs are not encoded and the quantized DCT coefficients are taken from the FB during VCR operations. In other words, the data in these MBs are shared among the FB and SRB. Therefore, the storage requirement of the SRB can be reduced remarkably.

TABLE 1

PERCENTAGE OF SMB FOR VARIOUS SEQUENCES.

Claire	91.33%
Grandma	86.78%
Salesman	64.20%
Carphone	59.27%
Table Tennis	51.93%
Foreman	45.03%
Football	39.40%

For a real world image sequence, the block motion field is usually gentle and smooth, and varies slowly. As a consequence, the distribution of motion vector is center-biased [10], as demonstrated by some typical examples as shown in Table 1 which shows the distribution of SMB for various sequences, including “Claire”, “Grandma”, “Salesman”, “Carphone”, “Foreman”, “Table Tennis” and “Football”. These sequences have been selected to emphasize different amount of motion activities. It is clear that over 90% and 39% of the MBs are SMB for sequences containing a low and high amount of motion activities respectively. The more SMBs in the SRB, the more bits can be saved.

4. EXPERIMENTAL RESULTS

In this section, we show a large amount of experimental works in order to evaluate the performance of the proposed SRB when applied to the dual-bitstream streaming system with VCR support. MPEG-4 encoder [11] was employed to encode various video sequences with different spatial resolutions and motion characteristics. “Foreman”, “Carphone”, “Claire” and “Grandma” are typical videophone sequences in QCIF (*176×144* pixels) format. “Salesman”, “Football”, and “Table Tennis” are in either CIF (*352×288* pixels) format or SIF (*352×240* pixels) format. Each test sequence was encoded into two bitstreams, the FB and the SRB (or RB). For all test sequences, the frame-rate of the video stream was *30* frames/s and the GOP length was 14 with an I-P structure.

In Table 2, we show the bitstream size and the average PSNR value for each test sequence that was encoded into the RB and SRB. In this table, ΔPSNR and ΔSIZE represent a PSNR change and percentage change in the bitstream of the SRB when compared to the original RB. A positive value means an increment whereas a negative value means a decrement. It can easily be seen that the required storage in the server of the proposed SRB is much fewer that that of the original RB. The results are more significant for the sequences “Salesman”, “Claire”, and “Grandma” as shown in Table 2. In these sequences, the size of the SRB can be reduced by 40-48% as compared to the original RB. It is due to the reason that these sequences contain more SMBs in which the redundancy to be exploited between the two bitstreams becomes more significant. For sequences containing high motion activities such as “Football”, “Table Tennis”,

TABLE 2

AVERAGE PSNR AND BITSTREAM SIZE FOR VARIOUS SEQUENCES.

SEQUENCES	RESOLUTION	BIT RATE OF FB	FB		RB		SRB		ΔPSNR (dB)	ΔSIZE
			PSNR (dB)	SIZE (KB)	PSNR (dB)	SIZE (KB)	PSNR (dB)	SIZE (KB)		
Salesman	*(352×288)*	3 Mbps	42.171	2407.074	40.600	2498.703	40.265	1292.170	-0.334	-48.29%
Football	*(352×240)*	3 Mbps	34.318	2554.604	32.825	2781.006	32.705	2144.181	-0.119	-22.90%
TableTennis	*(352×240)*	3 Mbps	38.907	2248.761	37.094	2362.411	36.644	1786.607	-0.449	-24.37%
Foreman	*(176×144)*	128 Kbps	29.019	107.365	27.682	121.813	27.541	96.051	-0.141	-21.15%
Carphone	*(176×144)*	128 Kbps	33.036	107.729	31.799	119.139	31.643	86.188	-0.156	-27.66%
Claire	*(176×144)*	128 Kbps	40.673	103.487	39.973	110.903	39.648	66.702	-0.325	-39.86%
Grandma	*(176×144)*	128 Kbps	36.322	107.430	35.843	116.936	35.726	68.207	-0.118	-41.67%

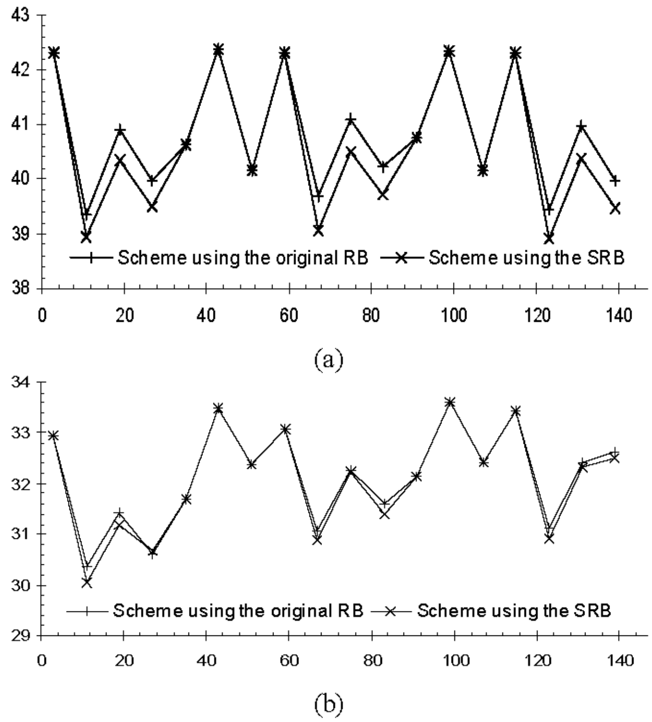

Fig. 3. PSNR performances by using the original RB and the SRB in the fast-backward mode with a speed-up factor of 8 for the (a) "Salesman", and (b) "Carphone" sequences.

"Foreman" and "Carphone", there still have good savings, as tabulated in Table 2.

The average PSNR values of the RB and SRB are also shown in Table 2. They show that the average PSNR values will slightly degraded by about 0.118 dB to 0.449 dB for the SRB. The degradation is due to the approximation in (5) and (6). In fact, this small degradation reflects the quality of the reconstructed frames during backward playback. In other VCR operations, the quality degradation is also negligible as shown in Fig. 3. It illustrates that the PSNR comparison for decoding the requested frames by using the RB and SRB when the fast backward operation with a speed-up factor of 8 is issued at the end of the sequence. In this figure, the "Salesman" and "Carphone" sequences were encoded at 3.0Mbits/s and 128Kbits/s, respectively. When the server performs bitstream switching by using the proposed SRB, there is a slight PSNR drop in some requested frames. However, this negligible degradation is not significant visually in the fast backward modes since the fast display speed will mask out most of the spatial distortion. Therefore, it can easily be seen that the proposed SRB in the dual-bitstream scheme can achieve similar performance to the scheme using the RB during various VCR operations.

5. CONCLUSIONS

An efficient technique for reducing storage requirement of the server has been proposed in this paper. It can eliminate the possible redundancy between the dual bitstreams. The proposed technique can exploit a large amount of non-MC MBs existed in real-world video sequences. With the motion information, the video streaming server classifies some MBs as skipped MBs (SMBs). A SMB is the one that the information about the MB is not necessary to be stored in the server and it is taken directly from the FB. By sharing the data between the dual bitstreams, a new and simplified RB (SRB) is used instead of the RB in the dual-bitstream scheme. Simulation results show that, with our proposed SRB, the dual-bitstream scheme reduces the storage requirement of the server with only a slight drop in PSNR for various browsing operations

6. ACKNOWLEDGMENTS

The work described in this paper is partially supported by the Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, Hong Kong Polytechnic University and a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (PolyU 5216/04E). Tak-Piu Ip acknowledges the research studentships provided by the University.

7. REFERENCES

[1] Video Coding for Low Bitrate Communication, ITU-T Recommendation H.263, May 1997.
[2] ISO/IEC 13818-2, "Information Technology - Generic Coding of Moving Pictures and Associated Audio Information: Video," 1996.
[3] ISO/IEC 14496-2, "Information Technology - Coding of audio-visual Objects – Part 2: Video," 2004.
[4] R. Srinivasan and K. R. Rao, "Predictive coding based on efficient motion estimation," IEEE Transactions on Communications, Vol. 33, No. 9, pp. 1011-1015, September 1985.
[5] Y. L. Chan and W. C. Siu, "An efficient search strategy for block motion estimation using image features", IEEE Transactions on Image Processing, Vol. 10, No. 8, pp. 1223-1238, August 2001.
[6] S. J. Wee, "Reversing motion vector fields," in Proc. IEEE International on Conference Image Processing 1998 (ICIP98), pp. 209-212, October 1998.
[7] C. H. Fu, Y. L. Chan and W. C. Siu, "Fast motion estimation and mode decision for H.264 reverse transcoding," Electronics Letters, Vol. 42, No. 24, November 2006, pp. 1385-1386.
[8] C. H. Fu, Y. L. Chan, and W. C. Siu, "Efficient reverse-play algorithms for MPEG video with VCR support," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 16, No.1, pp.19-30, January 2006.
[9] C. W. Lin, J. Zhou, J. Youn, and M. T. Sun, "MPEG video streaming with VCR functionality," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, No. 3, pp. 415-425, March 2001.
[10] J.Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," IEEE Trans. Circuits Syst. for Video Technol., Vol. 8, pp. 369-377, Aug. 1998.
[11] ISO/IEC JTC1/SC29/WG11 N3908, "MPEG-4 Video Verification Model Version 18.0," 2001.