Brief Papers

Redundancy Reduction Technique for Dual-Bitstream MPEG Video Streaming With VCR Functionalities

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

***Abstract*—With the proliferation of online video contents, it is highly desirable that video streaming systems are able to provide fast and effective video browsing. However, the predictive coding techniques adopted in current compression standards such as MPEG severely complicate these browsing operations. One approach to achieve browsing functionalities is to store an additional reverse-encoded bitstream into the server. Unfortunately, this extra bitstream approximately doubles the storage requirement of the video server. In this paper, we make use of 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. A novel macroblock-selection strategy is then proposed in the server to access and manipulate 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 alleviates the storage increase due to the additional reverse-encoded bitstream.**

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

I. INTRODUCTION

VIDEO streaming applications over the Internet are gaining popularity in recent years, mainly due to the emergence of efficient video compression and broadband networking technologies. Streaming applications such as video-on-demand allow users to ubiquitously access and retrieve various videos over the networks by using software players or digital set-top box devices. However, 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 of video browsing. For example, the video playback devices usually have limited fast-forward/backward playback flexibilities, or even they cannot provide backward playback. The 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 are then not invariant to changes in frame order and this fact makes some video cassette recording (VCR) operations complicated. Consider, for example, an MPEG encoded sequence with a simple I-P structure. If the requested frame is an I-frame, the server only needs to send this frame, and the decoder can decode it immediately. However, if the requested frame is a P-frame, the server needs to send all P-frames from the previous nearest I-frame to form this requested frame.

There have been many techniques proposed in the literature, that handle the problem of VCR operations. These techniques can roughly be categorized into two groups by whether the server needs to store additional bitstreams. The first category refers to the techniques without storing additional information in the server. They are designed to manipulate the original encoded bitstream in order to achieve VCR operations. In [6], [7], a backward-play transcoder was described to convert a sequence with I-/P-frames into another I–P bitstream with reverse frame order. Although some methods of estimating the reverse motion vectors for the new I–P bitstream based on the forward motion vectors of the original I–P bitstream were used to reduce the computational complexity of this transcoding process, they still require much computation. To avoid the transcoding complexity, we also suggested some compressed-domain techniques [8] by exploring the motion relationship between two adjacent frames in the server to provide backward playback over a network. This scheme can successfully reduce the required network bandwidth and the decoder effort of VCR requests. Since there is no inter-frame prediction between the last frame of a Group-of-Pictures (GOP) and the first frame of its successive GOP, the motion relationship disappears and thus this technique cannot be applied to GOP boundaries. In the instant of backward playback across GOP boundaries, the required complexity of the decoder and the required bandwidth of the network increase inevitably. Furthermore, the scheme in [8] is only applicable to backward playback. Huang *et al.* [9] investigated a possible binary tree GOP structure so that the transmission overhead caused by frame dependencies can be reduced during video browsing. Since this approach modifies the original bitstream, it affects the coding efficiency of forward playback. In the second category, additional bitstreams are required to assist the VCR operations. In [10], a client-server based video streaming architecture was proposed to allow discrete speed-up granularity by storing a separate pre-encoded video bitstream for each speed-up factor. The bitstream with

---

Manuscript received January 12, 2007; revised February 25, 2008. This work is supported in part by the Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, Hong Kong Polytechnic University and in part by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (PolyU 5125/06E). Tak-Piu Ip acknowledges the research studentships provided by the University.

The authors are with the Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong (e-mail: itbill@gmail.com; enylchan@polyu.edu.hk; enwcsiu@polyu.edu.hk).

Digital Object Identifier 10.1109/TBC.2008.2000289

suitable temporal resolution, which works according to the user's request, is then sent. However, the speed-up granularity is limited by the number of pre-encoded bitstreams and it forces all users to rely on the author's judgment as to speed-up granularity. Besides, additional storage is unavoidable in the server as the number of pre-encoded bitstreams increases. Liu *et al.* [11] 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 factor. This dual-bitstream scheme can alleviate the decoder complexity while maintaining the low network bandwidth requirement on VCR operations. However, extra storage is required to store the extra RB. The contribution of this paper is to reduce the size of the RB in the dual-bitstream scheme. In the proposed scheme, a simplified RB is designed to reuse some macroblock (MB) data from the FB by exploiting their redundancy. We then propose a novel MB-selection strategy to adaptively select the appropriate MBs from the two bitstreams. Note that this proposed scheme belongs to the second category.

The organization of this paper is as follows. Section II of this paper reviews the dual-bitstream scheme. The proposed simplified RB and MB-selection strategy used in the dual-bitstream video streaming system are then described in Sections III and IV, respectively. Experimental results are presented in Section V. Finally, some concluding remarks are provided in Section VI.

II. DUAL-BITSTREAM MPEG VIDEO STREAMING SYSTEM

In order to facilitate different VCR trick modes of the MPEG video streaming system, a dual-bitstream technique was proposed in [11] which 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 sequence in reverse order. Fig. 1 shows an illustrative example of the FB and RB in which the sequence is coded in I-/P-frames with a GOP size of 14 frames. Note that I-frames in the RB are interleaved between I-frames in the FB. Since B-frames are not used as references for later frames, and are not needed to be sent over the network or decoded by the decoder, for the sake of simplicity, we focus our discussions on the case that the video stream contains I- and P-frames only. With the help of RB, when the client requests the backward-play mode, the server will send the bits from the RB. After the server receives requests on fast-forward/backward and random access modes, the server uses the frame-selection strategy to determine which frames in either the FB or RB should be transmitted to the client by minimizing the cost using bitstream switching. If the requested frame is a P-frame, the frame-selection strategy measures (i) the cost of decoding the next requested P-frame from the current displayed frame $(cost_C)$, (ii) the cost of decoding the next requested P-frame from the nearest I-frame in the FB $(cost_{FB})$, and (iii) the cost of decoding the next requested P-frame from the nearest I-frame in the RB $(cost_{RB})$. In [11], the "cost" means the number of required frames to be sent over the network. Based on $cost_C$, $cost_{FB}$, and $cost_{RB}$, the least cost should be selected to initiate the decoding. The frame-selection strategy may switch from the FB to the RB and vice versa. In other words, it determines which bitstream should be selected next and its decoding direction. To illustrate the scheme, let us make use of the structure of dual bitstreams in Fig. 1. Assume that 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, $cost_C$, $cost_{FB}$, and $cost_{RB}$ are computed to determine whether the current displayed frame (frame 14 of the FB), the nearest I-frame in the FB (again, frame 14 of the FB) or the nearest I-frame in the RB (frame 7 of the RB) is selected to initiate the decoding of the requested frames. In this example, $cost_C$, $cost_{FB}$, and $cost_{RB}$ are equal to 6, 6, and 2 respectively. Note that $cost_C$ and $cost_{FB}$ are equal since the current displayed frame and the nearest I-frame in the FB are the same in this example. Fig. 1 also shows the actual frames required to be sent using different costs in details. Hence, frame 8 will be decoded from frame 7 of the RB (an I-frame) since $cost_{RB}$ has the least cost. It implies 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 this will last until the next I-frame. However, the subjective degradation is not visually significant due to the fast changes of the video content displayed [11].

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

Although the dual-bitstream scheme can provide an effective way to support VCR operations for MPEG video, it requires additional storage for the RB. A technique for reducing the storage requirement of the RB is considered in this paper. The proposed technique attempts to exploit redundancy in some MBs found between the two bitstreams. The situation in MB level of the dual-bitstreams is depicted in Fig. 2. In the server, $MB^{FB_n}_{(k,l)}$ and $MB^{RB_n}_{(k,l)}$ 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, $e^{FB_n}_{(k,l)}$, and the RB, $e^{RB_{n-1}}_{(k,l)}$, are given by

$$e^{FB_n}_{(k,l)} = MB^{FB_n}_{(k,l)} - MCMB^{FB_{n-1}}\left(mv^{FB_n}_{(k,l)}\right) \qquad (1)$$

and

$$e^{RB_{n-1}}_{(k,l)} = MB^{RB_{n-1}}_{(k,l)} - MCMB^{RB_n}\left(mv^{RB_{n-1}}_{(k,l)}\right) \qquad (2)$$

where $MCMB^{FB_{n-1}}(mv^{FB_n}_{(k,l)})$ stands for the motion-compensated MB of $MB^{FB_n}_{(k,l)}$ which is translated by the motion vector $mv^{FB_n}_{(k,l)}$ in frame $n-1$ of the FB and $MCMB^{RB_n}(mv^{RB_{n-1}}_{(k,l)})$ represents the motion-compensated MB of $MB^{RB_{n-1}}_{(k,l)}$ with the displacement of the motion vector $mv^{RB_{n-1}}_{(k,l)}$ in frame n of the RB. Note that, in contrast to the FB, frame $n-1$ is predicted

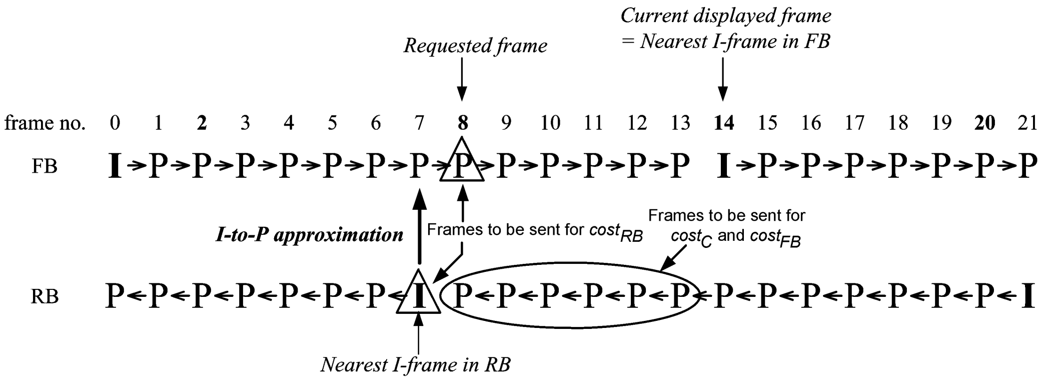


Fig. 1. The dual-bitstream scheme: fast-backward operation with a speed-up ratio of 6.

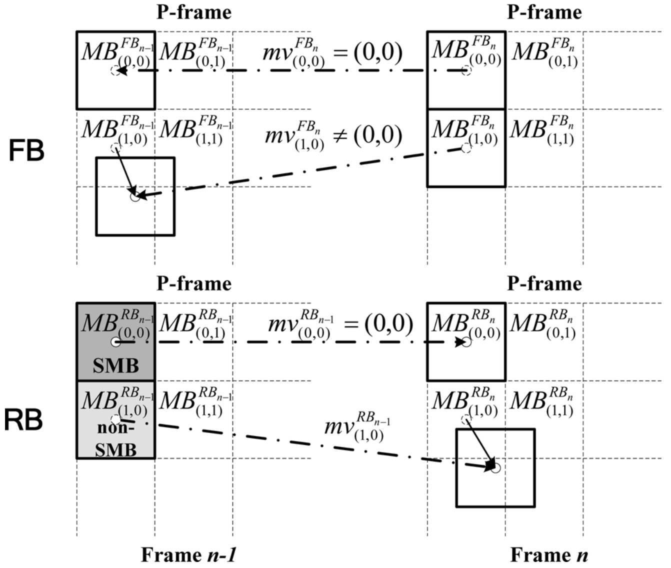


Fig. 2. MB views of the proposed dual-bitstream and the definition of the SMB and non-SMB.

from frame n in the RB since this bitstream is generated by encoding the video frames in reverse order. All the 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, decoded 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 that 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 as possible in the FB. To do this, 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 as possible in the FB for encoding $MB_{(k,l)}^{RB_{n-1}}$ in 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

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

As discussed before, the reconstructed pixels of frame n in the FB and RB are perceptually similar to each other since they encode the same video content. The only discrepancy is that they use different reference frames in opposite directions. 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,

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

Similarly,

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

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

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

From (3) and (7), we get

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

TABLE I
PERCENTAGE OF SMB FOR VARIOUS SEQUENCES

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

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,

$$DCT\left(e_{(k,l)}^{RB_{n-1}}\right) \approx -DCT\left(e_{(k,l)}^{FB_n}\right) \tag{9}$$

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

$$Q\left[DCT\left(e_{(k,l)}^{RB_{n-1}}\right)\right] \approx -Q\left[DCT\left(e_{(k,l)}^{FB_n}\right)\right] \tag{10}$$

$Q[DCT(e_{(k,l)}^{RB_{n-1}})]$ 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. The 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.

For a real world image sequence, the block motion field is usually gentle, smooth, and varies slowly. As a consequence, the distribution of motion vector is center-biased [12], as demonstrated by some typical examples as shown in Table I which shows the percentage 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. The percentage of SMB is calculated according to the following equation.

$$\%\,of\,SMB = \frac{Total\;no.\;of\;SMB\;in\;SRB}{Total\;no.\;of\;MB\;in\;SRB} \tag{11}$$

In Table I, 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.

IV. ARCHITECTURE OF VIDEO STEAMING SERVER WITH THE SUPPORT OF SRB

The above section only addresses how to eliminate the redundancy between the dual bitstreams. When a VCR operation requests a particular frame from the SRB, the server adopts a MB-selection strategy which needs to extract the appropriate MBs from the FB in order to insert the shared data into the SRB prior to transmission. Fig. 3 shows the proposed architecture of a video streaming server. When a frame from the SRB is requested, the appropriate motion vectors are extracted by the motion vector extractor from the FB. Based on the motion vectors, the MB classifier arranges the MBs from the SRB into two types: a non-SMB and a SMB. The server does not need to do anything for non-SMBs in the SRB, and switch SW is connected to position B. For a SMB, the server uses only the data from the FB to reconstruct the requested MB in the SRB by switching SW to position A. In each SMB, the corresponding VLC codewords are extracted from the corresponding MB in the FB. Afterwards, these VLC codewords are undergone VLC decoding to reconstruct the quantized DCT coefficients. From (10), the signs of all these quantized DCT coefficients are inverted to form the desired coefficients through the sign inverter, as shown in Fig. 3. The sign inverted coefficients are then encoded to its final VLC codewords for the SMB. The bitstream re-generator integrates these new VLC codewords with the non-SMBs in the SRB before transmitting to the network. Note that the server only needs to perform variable length decoding and encoding, and a complete decoding and encoding are not required at the server. It only causes a slightly increase in the server complexity.

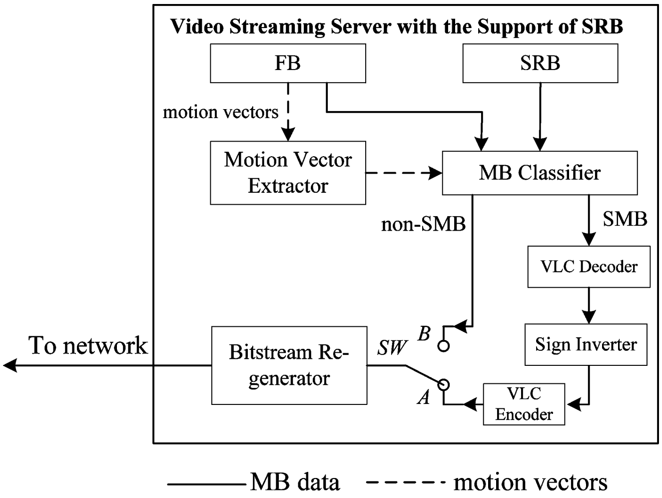


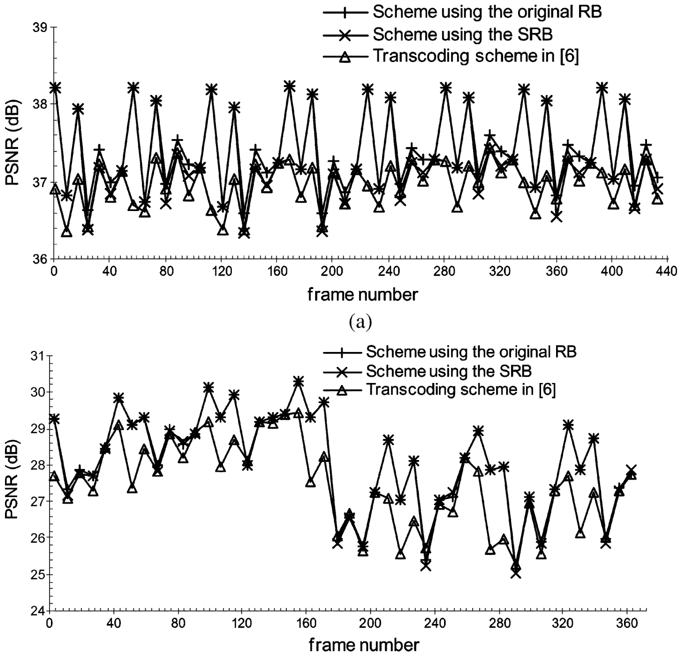
Fig. 3. The proposed architecture for the dual-bitstream video streaming scheme with VCR functionality.

V. EXPERIMENTAL RESULTS

Many experiments have been conducted to evaluate the performance of the proposed SRB when applied to the dual-bitstream streaming system with VCR support. An MPEG-4 encoder [13] 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. All sequences were encoded at two different bit rates. Each test sequence was encoded into two bitstreams, the FB and the SRB (or RB), and I-frames in the SRB (or RB) are interleaved between I-frames in the FB. The SRB (or RB) can be obtained

TABLE II
AVERAGE PSNR AND BITSTREAM SIZE FOR VARIOUS SEQUENCES

Sequences	Bit rate of FB	FB PSNR (dB)	FB Size (KB)	RB PSNR (dB)	RB Size (KB)	SRB PSNR (dB)	SRB Size (KB)	ΔPSNR (dB)	ΔSize
Salesman	3 Mbps	42.171	2407.074	40.600	2498.703	40.265	1292.170	-0.334	-48.29%
(*352×288*)	1.5 Mbps	37.690	1087.164	36.842	1157.830	36.711	687.312	-0.131	-40.64%
Football	3 Mbps	34.318	2554.604	32.825	2781.006	32.705	2144.181	-0.119	-22.90%
(*352×240*)	1.5 Mbps	30.653	1168.521	28.992	1346.715	28.876	1118.515	-0.116	-16.94%
TableTennis	3 Mbps	38.907	2248.761	37.094	2362.411	36.644	1786.607	-0.449	-24.37%
(*352×240*)	1.5 Mbps	33.927	1149.868	32.693	1263.781	32.377	1000.946	-0.316	-20.80%
Foreman	128 Kbps	29.019	107.365	27.682	121.813	27.541	96.051	-0.141	-21.15%
(*176×144*)	64 Kbps	26.204	56.651	25.324	60.575	25.239	51.715	-0.084	-14.63%
Carphone	128 Kbps	33.036	107.729	31.799	119.139	31.643	86.188	-0.156	-27.66%
(*176×144*)	64 Kbps	29.416	58.930	28.474	63.978	28.352	51.616	-0.122	-19.32%
Claire	128 Kbps	40.673	103.487	39.973	110.903	39.648	66.702	-0.325	-39.86%
(*176×144*)	64 Kbps	34.496	54.969	33.772	58.391	33.553	41.635	-0.219	-28.70%
Grandma	128 Kbps	36.322	107.430	35.843	116.936	35.726	68.207	-0.118	-41.67%
(*176×144*)	64 Kbps	32.350	55.304	31.944	57.712	31.877	41.461	-0.067	-28.16%



(a)

(b)

Fig. 4. 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.

TABLE III
AVERAGE PSNR OF ALL POSSIBLE REQUESTED FRAMES WITH RESPECT TO ALL STARTING POINT FOR VARIOUS SEQUENCES

Sequences	Bit rate of FB	RB (dB)	SRB (dB)	ΔPSNR (dB)
Salesman	3 Mbps	40.308	40.271	-0.037
(*352×288*)	1.5 Mbps	36.703	36.679	-0.024
Football	3 Mbps	32.311	32.329	0.018
(*352×240*)	1.5 Mbps	28.426	28.436	0.010
TableTennis	3 Mbps	36.654	36.517	-0.137
(*352×240*)	1.5 Mbps	32.332	32.235	-0.097
Foreman	128 Kbps	27.503	27.535	0.032
(*176×144*)	64 Kbps	25.351	25.360	0.009
Carphone	128 Kbps	31.688	31.706	0.018
(*176×144*)	64 Kbps	28.483	28.482	-0.001
Claire	128 Kbps	40.000	39.978	-0.022
(*176×144*)	64 Kbps	33.910	33.886	-0.024
Grandma	128 Kbps	35.905	35.897	-0.008
(*176×144*)	64 Kbps	32.047	32.050	0.003

by re-encoding the FB in reverse order. Note that the generation of the SRB (or RB) is done offline. 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 II, we show the bitstream size and the average PSNR value for each test sequence that was encoded into the RB and SRB at two different bit rates. In this table, $\Delta\mathrm{PSNR}$ and $\Delta\mathrm{SIZE}$ 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 than that of the original RB in both bit rates. The results are more significant for the sequences “Salesman”, “Claire”, and “Grandma” as shown in Table II. In these sequences, the size of the SRB can be reduced by 40–48% and 28–40% as compared to the original RB at high bit rate and low bit rate, respectively. 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”, “Foreman”, and “Carphone”, there are still good savings, as tabulated in Table II. Besides, this table signifies that the size of the SRB can be reduced more remarkably for sequences encoded at high bit rate. The reason behind is that, at low bit rate, a considerable percentage of DCT blocks has a significant amount of zero elements in MBs of the original RB. In these MBs, the encoder allocates fewer bits to encode the residuals. If those MBs are considered as SMBs in the proposed SRB, it cannot achieve as much saving of bits as the case in video sequences encoded at high bit rate.

The average PSNR values of the RB and SRB are also shown in Table II. They show that the average PSNR values will slightly degraded by about 0.067 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

TABLE IV
AVERAGE PSNR AND BITSTREAM SIZE FOR VARIOUS SEQUENCES WITH AN IBBP STRUCTURE

	FB		**RB**		**SRB**			
Sequences	**PSNR (dB)**	**Size (KB)**	**PSNR (dB)**	**Size (KB)**	**PSNR (dB)**	**Size (KB)**	**ΔPSNR (dB)**	**ΔSize**
Salesman	42.398	1677.772	40.766	1734.381	40.449	575.826	-0.317	-66.80%
(*352×288*)	38.923	921.324	37.541	993.411	37.438	373.368	-0.103	-62.42%
Football	33.018	1994.129	31.698	2065.088	31.537	782.374	-0.161	-62.11%
(*352×240*)	29.553	999.578	28.022	1080.411	27.874	451.586	-0.148	-58.20%
TableTennis	37.791	1818.426	35.955	1914.584	35.393	703.214	-0.562	-63.27%
(*352×240*)	32.832	887.605	31.722	944.484	31.345	392.306	-0.377	-58.46%
Foreman	29.131	86.722	27.580	100.195	27.489	49.690	-0.091	-50.41%
(*176×144*)	26.329	50.581	25.015	55.872	24.976	24.964	-0.039	-55.32%
Carphone	32.528	86.608	31.208	93.533	31.074	37.273	-0.134	-60.15%
(*176×144*)	29.200	51.533	28.075	54.719	28.001	21.411	-0.074	-60.87%
Claire	40.200	76.120	38.831	80.224	38.564	33.156	-0.267	-58.67%
(*176×144*)	34.357	42.030	33.350	43.873	33.097	18.165	-0.253	-58.60%
Grandma	36.092	78.631	35.270	82.238	35.165	32.773	-0.105	-60.15%
(*176×144*)	32.301	43.665	31.729	44.915	31.663	17.737	-0.066	-60.51%

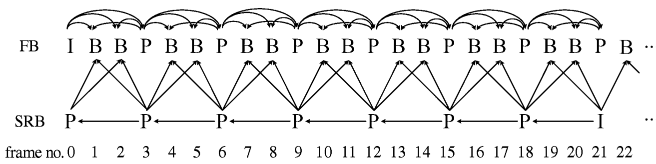


Fig. 5. The prediction structure of SRB with B-frames.

operations, the quality degradation is also negligible as shown in Fig. 4. 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 sequences. In this figure, the "Salesman" and "Carphone" sequences were encoded at 1.5 Mbits/s and 64 Kbits/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 visually significant in the fast backward modes since the fast display speed will mask out most of the spatial distortion. Note that the transcoding scheme based on [6] is also shown in Fig. 4 for a comparison. Although this scheme does not need to store an extra bitstream in the server, the re-encoding process results in serious drop in PSNR, as depicted in Fig. 4. Besides, the re-encoding process induces heavy computational requirement in the server. We have also done exhaustive simulation on the PSNR performances of all possible combinations of requested frames and start frames (the frame in which a user issues a VCR operation). Table III lists out the average PSNR values of these possible combinations when the sequences are coded at high bit rate and low bit rate. In fact, these values indicate the average PSNR performance of the random access mode for using the RB and SRB. Random access is an important operation for providing VCR capability in a video browsing system. From the statistics as shown in Table III, it can easily be seen that the proposed SRB in the dual-bitstream scheme can achieve almost the same quality as compared to the scheme using the RB during random access. Therefore, the SRB can reduce the storage requirement significantly as well as keeping the reconstruction quality of various VCR browsing operations.

In addition, we have also simulated the performance of the proposed SRB with the structure "IBBPBBPBBP…" as the GOP. To further reduce the redundancy between the SRB and the FB, one straightforward approach is to exploit the symmetry of the B-frame prediction process adopted in the MPEG bitstream. In the FB as shown in Fig. 5, each B-frame is predicted from two reference frames (the immediately preceding I- or P-frame and the immediately succeeding I- or P-frame) using the backward and forward motion vectors. The coding structure of the SRB is basically to reverse the order of the frames, and this will result in a swapping of the backward and forward reference frames from the FB. Therefore, the SRB can re-use the B-frames in the FB by simply swapping its backward and forward motion vectors, as depicted in Fig. 5. In other words, when accessing a B-frame in the SRB, the server only needs to VLC decode the B-frame in the FB, to swap the backward and forward motion vectors, and to VLC re-encode the B-frame. The processed data are then combined into the SRB. This enhanced SRB provides a further saving as shown in Table IV, in which two B-frames are added between two anchor I-/P-frames. This further demonstrates the effect of the proposed SRB when the bitstream contains B-frames.

VI. CONCLUSIONS

In this paper, an efficient technique for reducing storage requirement of a server has been proposed to eliminate the possible redundancy between the dual bitstreams. The proposed technique exploits 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 just a slight drop in PSNR for various VCR operations. The PSNR drop is due to a mismatch between MBs in the FB and the SRB. The subjective degradation observed is relatively insensitive to human eyes because the

VCR operation always lasts for a few frames or the fast display speed could mask out most of the distortions. However, in some professional applications such as studio editing, it may still be desirable to reduce the mismatch. One of our future works could focus on techniques to reduce the mismatch. For instance, the SMBs are now not encoded and the quantized DCT coefficients are directly taken from the FB during VCR operations. Hence a possible future investigation is to find an efficient way to add drift-compensated data in SMBs, the approach of which could be based upon the decoded MBs from the forward and backward bitstreams.

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 Trans. Communications*, vol. COM-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 Trans. 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)*, October 1998, pp. 209–212.

[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, pp. 1385–1386, November 2006.

[8] C. H. Fu, Y. L. Chan, and W. C. Siu, "Efficient reverse-play algorithms for MPEG video with VCR support," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 16, no. 1, pp. 19–30, January 2006.

[9] C. H. Huang, K. C. Yang, and J. S. Wang, "A low-cost unrestricted fast playback scheme for video streaming," *IEEE Trans. Circuits and Systems-II: Express Briefs*, vol. 9, no. 7, pp. 384–388, July 2005, Col. 52.

[10] N. Omoigui, L. He, A. Gupta, J. Grudin, and E. Sanocki, "Time-compression: System concerns, usage, and the benefits," in *Proceedings of ACM SIGHI Conference*, May 1999, pp. 136–143.

[11] C. W. Lin, J. Zhou, J. Youn, and M. T. Sun, "MPEG video streaming with VCR functionality," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 415–425, March 2001.

[12] 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.

[13] IS0/IEC JTC1/SC29/WG11 N3908, "MPEG-4 Video Verification Model Version 18.0," 2001.