A New Motion Vector Composition Algorithm for Fast-forward Video Playback in H.264

Tsz-Kwan Lee, Chang-Hong Fu, Yui-Lam Chan, and Wan-Chi Siu
Centre for Signal Processing, Department of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

Abstract—With the rapid growth of streaming digital videos, it is desirable to access video segments of interest by searching through the video contents with a faster speed than a normal playback. Fast-forward playback is the key function that enables quick browsing of videos. It can be realized by a frame-skipping transcoding approach which transcodes only the frames required for playback at the desired fast speed. Various motion vector (MV) composition algorithms aim at reducing the computational complexity of the transcoder. They only perform fairly in limited skipped frames scenarios. In this paper, a new vector selection algorithm is proposed to compose a new motion vector (MV) from a set of candidate MVs for minimizing prediction errors due to a larger frame-skipping factor. Experimental results show that the proposed algorithm can deliver a remarkable improvement on the rate-distortion performance over other algorithms.

I. INTRODUCTION

Current video compression standards are basically developed for the purposes of efficient video storage and transmission, not browsing. This limitation is due to the use of motion compensated prediction for compression. As a result, recent video playback devices only offer limited numbers of controls for video browsing because motion compensation heavily utilizes the frame dependency and severely complicates the fast-forward/backward operations, which are the major functions for video browsing [1], [2]. Thus, fast-forward playback function requires more network bandwidth to send all the related frames in addition to the actual requested frames [3]–[5]. Besides, it also requires high computational capability in clients’ decoders to decode all these extra frames. One simple approach to implement the fast forward/backward playback is just to send and decode the I-frames. However, if the video applications involve an encoded bitstream with a very large GOP size or require high-precision in video-frame access, sending only I-frames may not be an acceptable solution.

Some approaches have been proposed for the realization of fast playback. In [1], [3], [4], various dual-bitstream schemes were proposed to store the forward/backward-encoded bitstreams in the server for frame selection. Nevertheless, it approximately doubles the storage requirement for the server. To tackle this, a frame-skipping transcoding approach for fast-forward playback was proposed in [2], [5]. Transcoding begins with full video decoding in the server, selects and re-encodes the frames required for fast playback at a desired speed, and sends the transcoded bitstream to the clients for decoding and display. It induces high computational complexity in the re-encoding process. Hence, some MV composition methods are used to expedite the re-encoding process. The forward dominant vector selection (FDVS) [6] is suggested to be the best in fast-forward frame-skipping transcoding [5]. However, FDVS does not work well for a large speed-up factor in fast video playback since the composition of new MVs may not represent the current macroblock (MB) anymore. In this case, the quality of the transcoded videos deteriorates.

In this paper, we propose a more reliable algorithm to compose new MVs with a large speed-up factor. The MV composition is based on the relevant area of the current MB. It can also track several possible candidates related to the current MB and select the best candidate in transcoding. The organization of this paper is as follows. In Section II, we discuss the impacts of a large speed-up factor on the performance of FDVS. Section III describes our proposed algorithm for frame-skipping transcoding. Simulation results are presented in Section IV. Finally, some concluding remarks are provided in Section V.

II. PROBLEMS OF USING FDVS

Fig. 1(a) illustrates an example of fast-forward transcoding using FDVS. Only four MBs are shown in each frame. We assume that $MB_n^k$ represents the $k$th MB in Frame $n$ with the MV $mv_{n-1}^k$ which uses Frame $n-1$ as the reference in Fig. 1(a). The fast-forward speed-up factor is 3 where Frame $n-1$ and Frame $n-2$ are skipped for fast playback, then $mv_{n-1}^k$ becomes not valid. It is necessary to find the new MV of $MB_n^k$ with Frame $n-3$ as the reference, i.e.

![Fig. 1. A working example of FDVS.](image-url)
$\text{mv}_{n-3}^k$ is shown in dotted arrow of Fig. 1(a). For each MB, FDVS selects one dominant MV carried by a dominant MB which has the largest overlapping segment with the motion-compensated MB of $MB_n^1$ in the previous frame. Considering the motion-compensated MB of $MB_n^1$ overlaps with four MBs, $MB_{n-1}^1$, $MB_{n-2}^1$, $MB_{n-3}^1$ and $MB_{n-4}^1$, in Frame $n-1$ of Fig. 1(a), $MB_{n-1}^2$ is selected as the dominant MB while its MV $\text{mv}_{n-1}^2$ is the dominant MV. This dominant vector selection process is repeated until the non-skipped frame is reached, i.e. Frame $n-3$ in this example. Therefore, $\text{mv}_{n-3}^1$ is composed by summing up the selected dominant MVs and can be written as (1)

$$\text{mv}_{n-3}^1 = \text{mv}_{n-1}^1 + \text{mv}_{n-2}^3 + \text{mv}_{n-2}^2 - n-3 \tag{1}$$

FDVS can provide promising results for MV composition in transcoding and becomes the most popular technique compared with other existing algorithms, which aim to reduce the computational complexity of motion estimation processes in video encoders [5], [6]. However, FDVS does not work well for consecutively dropping a large number of frames, which is very common in fast-forward playback. This phenomenon can be explained as depicted in Fig. 1(b), which is redrawn from Fig. 1(a). Here, $MB_{n-1}^2$ is selected to be the dominant MB and the corresponding $\text{mv}_{n-2}^2$ is used to determine the dominant MB in Frame $n-2$. It is observed that only the shaded area of $MB_{n-1}^2$ is actually relevant to target $MB_n^1$. However, FDVS also utilizes the irrelevant non-shaded area in $MB_{n-1}^2$ to compute dominant MB in Frame $n-2$. The relevant area of $MB_n^1$ further diminishes for more skipped frames. The cross-hatch shaded area only occupies a very minor portion of $MB_{n-2}^1$ as shown in Fig. 1(b). It seriously affects the accuracy of the composed MVs since a large irrelevant area to the target $MB_n^1$ is used to decide the dominant MB in Frame $n-3$.

### III. PROPOSED VECTOR SELECTION ALGORITHM

Based on the above observations of the FDVS process in section II, two criteria are set for our proposed algorithm in transcoding. First, only the area relevant to the target MB should be contributed for dominant MB selection. Second, the area relevant to the target MB should be kept as large as possible during MV composition.

#### A. Only Using Relevant Area for Vector Selection

According to the first criterion, Fig. 2 shows an improved mechanism for FDVS. When $MB_{n-1}^2$ is chosen as the dominant MB in the first step of FDVS, only the shaded area with slash lines, which is the relevant region to the target $MB_n^1$, is used to decide the next dominant MB in Frame $n-2$. Note that $MB_{n-2}^4$ is selected which contrasts to the selection of original FDVS where $MB_{n-2}^4$ is picked. Again, if Frame $n-2$ is also dropped, only the cross-hatch shaded area in Frame $n-2$ is used to determine the next dominant area in Frame $n-3$. This mechanism ensures only relevant area of $MB_n^1$ is employed in the MV composition. From Fig. 2, the resultant MV $\text{mv}_{n-3}^1$ is different from the result obtained by using FDVS in (1), and can be formed as (2)

$$\text{mv}_{n-3}^1 = \text{mv}_{n-1}^1 + \text{mv}_{n-2}^3 + \text{mv}_{n-2}^2 - n-3 \tag{2}$$

#### B. Maximizing Relevant Area

1) **Merging Process:** The objective of the second criterion is to maximize the relevant area used for dominant MB selection. Other non-dominant areas in the skipped frames, but relevant to $MB_n^1$, can also be utilized to enhance the usage of relevant area in $MB_n^1$. The reason is that the largest overlapping segment with the motion-compensated MB of $MB_n^1$ sometimes may not be dominant enough compared to the second largest one as shown in Frame $n-1$ of Fig. 3(a). As a result, the relevant area may diminish after MV composition. In the example shown in Fig. 3(a), the cross-hatch shaded area in Frame $n-2$ for selecting the next dominant MB becomes very small so it decreases the reliability of the resultant MV. To fully utilize the relevant area in $MB_n^1$, the proposed algorithm considers the homogeneity of MVs as well, which is essential to enlarge the relevant area for MV composition. We reuse the example in Fig. 3(a), but $\text{mv}_{n-2}^2$ is now equal to $\text{mv}_{n-1}^1 - \text{n-2}$ as shown in Fig. 3(b). In this case, the shaded area overlapped with $MB_{n-1}^2$ and $MB_{n-1}^4$ could be combined, and this merging area is for deciding the next dominant MB in Frame $n-2$. The selected MB in Frame $n-2$ is $MB_{n-2}^4$ where the area relevant to $MB_n^1$ is larger and is more reliable to determine the dominant MB in Frame $n-3$, compared to the case of Fig. 3(a).

2) **Multiple-candidates:** The merging process is appropriate for areas with homogeneous motion and it is particularly true
for MBs in the background and inside the moving objects. At the object boundary of a video object, we suggest using more than one candidate MB in order to expand the area relevant to the target MB in the MV composition. Assume that $C_{n-1}^i$ is the $i$th candidate in Frame $n-1$ sorted by the area of the overlapping segment. In Fig. 4, two candidate MBs are used to compose the MV for each step. In Frame $n-1$, $C_{n-1}^1$ and $C_{n-1}^2$ are the largest and second largest overlapping segments with the motion-compensated MB of $MB_1^n$, respectively. Therefore, both $MB_2^n$ and $MB_4^n$ are used to determine the next dominant MBs in Frame $n-2$ because both shaded areas in $MB_2^n$ and $MB_4^n$ are relevant to $MB_1^n$.

From the top diagram of Fig. 4, four candidates ($C_{n-2}^1$, $C_{n-2}^2$, $C_{n-2}^3$, and $C_{n-2}^4$) contributed from the motion-compensated segment of $C_{n-1}^A$ are considered for the next step. In addition, $C_{n-2}^1$ contributed from the motion-compensated segment of $C_{n-1}^B$ is also regarded as one of the possible candidates, as depicted in the bottom diagram of Fig. 4. Since two candidates are used for each step, $C_{n-2}^1$ and $C_{n-2}^2$ are chosen as the largest and second largest overlapping segments with their corresponding MBs, $MB_2^{n-1}$ and $MB_4^{n-1}$ respectively. The top diagram of Fig. 4 shows the same procedure of the MV composition as illustrated in Fig. 3(a). The bottom diagram gives an alternative path to compose the new MV, which uses the second largest candidate MB in Frame $n-1$. From Fig. 4, we observe that the cross-hatch shaded area in Frame $n-2$ of the bottom diagram, which is relevant to $MB_2^n$, and is used to decide the dominant MB in Frame $n-2$, is larger than that of the top diagram. In other words, even though $C_{n-1}^A$ is the largest overlapping segment in the first skipped frame, Frame $n-1$, it cannot guarantee that it is still the largest overlapping segment in the next skipped frame, Frame $n-2$. The use of multiple-candidate MBs for each skipped frame can increase the possibility of keeping the MBs with a large relevant area to the target MB during the MV composition. Since only two frames are skipped in this working example, two candidates are sufficient enough for each skipped frame. When more frames are skipped, a larger number of possible candidates is necessary. Note that the number of candidates can be selected by users according to the number of skipped frames and the desired video quality.

IV. SIMULATION RESULTS

Simulations have been performed to evaluate the overall efficiency of various MV composition algorithms in fast-forward playback. The H.264 reference codec (JM9.2) was employed to pre-encode the test sequences of CIF format (352×288 pixels) with 200 frames, including “Salesman”, “Foreman”, “Mobile”, “Coastguard”, and “Tempete”, at 30 frames/s with a fixed quantization parameter. Their first frame was encoded as I-frame, while others were P-frames in which a full-search motion estimation algorithm with a search window of 31×31 pixels was used to determine the MVs in the pre-encoded videos. The pre-encoded videos were then transcoded to various fast-forward videos with speed-up factors of 3, 5, 7, and 9. All of the picture types and quantization parameter were preserved during transcoding. For comparison, the full-search motion estimation (FS), the forward dominant vector selection algorithm (FDVS) [6], the extended version of FDVS (E-FDVS) [7], and the proposed multiple-candidate vector selection (MCVS-C) were used to obtain the MVs of the transcoded videos. In MCVS-C, C represents the number of candidate MBs selected for each skipped frame. In our simulations, C was set to 2 and 4 represented by MCVS-2.
Table I lists the transcoding results with a speed-up factor of 5 on different test sequences. The PSNR result for each transcoded frame is computed by comparing each transcoded frame with its original, uncompressed frame. In the table, $\Delta$dB and $\Delta$bits represent a PSNR change and a percentage change in total bits respectively when compared to FS. The positive values mean increments whereas negative values mean decrements. It is observed that the bits to be generated for the proposed algorithm are much fewer than that of FDVS and E-FDVS, especially in MCVS-4. It is because our proposed algorithm utilizes only the relevant area of the target MB in the MV composition and multiple-candidate selection plays an important role to keep relevant area as large as possible across the skipped frames. Regarding the computational complexity of the proposed MCVS-C, the only major overhead is the SAD calculation required for the final selection of different resulted MVs from different candidates. These extra computations could be neglected, as compared with FS.

Table II shows results by different algorithms for transcoding the “Coastguard” sequence.

Table I lists the transcoding results with a speed-up factor of 5 for various speed-up factors for transcoding the “Coastguard” sequence.

<table>
<thead>
<tr>
<th>Speed-up factors</th>
<th>Full Search</th>
<th>FDVS</th>
<th>E-FDVS</th>
<th>MCVS-2</th>
<th>MCVS-4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bits</td>
<td>PSNR</td>
<td>Bits</td>
<td>PSNR</td>
<td>Bits</td>
</tr>
<tr>
<td>3</td>
<td>4208</td>
<td>34.16 &lt;t&gt;(+22.80%)&lt;/t&gt;</td>
<td>2932 &lt;t&gt;(+8.29%)&lt;/t&gt;</td>
<td>2878 &lt;t&gt;(+7.80%)&lt;/t&gt;</td>
<td>2524</td>
</tr>
<tr>
<td>5</td>
<td>5070</td>
<td>33.87</td>
<td>2337</td>
<td>33.61</td>
<td>2201</td>
</tr>
<tr>
<td>7</td>
<td>5933</td>
<td>33.04</td>
<td>2013</td>
<td>32.51</td>
<td>1830</td>
</tr>
<tr>
<td>9</td>
<td>6796</td>
<td>32.37</td>
<td>1757</td>
<td>31.93</td>
<td>1505</td>
</tr>
</tbody>
</table>

Table II shows results by different algorithms for transcoding the “Coastguard” sequence with speed-up factors of 3, 5, 7, and 9. It is observed that the performances of FDVS and E-FDVS get worse as the speed-up factor increases. MCVS-2 outperforms E-FDVS and FDVS in terms of both PSNR and total generated bits for the cases with large speed-up factors. MCVS-4 can provide further reduction in generated bits, as shown in Table II. For fast-forward playback at 9 times the normal speed, $\Delta$bits of MCVS-4 is reduced to 8.76% while it is 15.84% in E-FDVS and 22.8% in FDVS.

The total bits generated and average PSNR obtained by different MV selection algorithms for the “Coastguard” sequence with speed-up factors ranged from 2 to 9 were also plotted in Fig. 5 and Fig. 6, respectively. It is significant to note that gaps in both PSNR and generated bits between FS and MCVS-4 become narrower. From these statistics, we can conclude that the proposed MCVS can provide outstanding results, especially in the case of a large speed-up factor.

V. CONCLUSION

In this paper, we have proposed a novel MV composition algorithm for realizing fast-forward playback of a pre-encoded video by frame-skipping transcoding. Our proposed multiple-candidate vector selection (MCVS) algorithm fully makes use of relevant areas to the target MB, and it is beneficial to perform fast-forward playback with a large speed-up factor. Its performance verified experimentally in terms of both quality and bits is substantially better than that of FDVS and E-FDVS. Besides, the proposed MCVS is adaptive in nature, and the number of candidate MBs can be adjusted according to the speed-up factor.

ACKNOWLEDGMENT

The work described in this paper is partially supported by the Centre for Signal Processing, Department of EIE, PolyU and a grant from the Research Grants Council of the HKSAR, China (PolyU 5120/07E). Tsz-Kwan Lee acknowledges the research studentship provided by the University.

REFERENCES