AN ARBITRARY FRAME-SKIPPING VIDEO TRANSCODER

Vasant Patil and Rajeev Kumar

Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur, WB 721 302, INDIA
e-mail: {vasantp, rkumar}@cse.iitkgp.ernet.in

ABSTRACT

In video transcoding, pre-encoded frames may be arbitrarily dropped to freely adjust the video to meet the network and client requirements. Since transcoding is carried out in real-time, incoming motion vectors are reused to reduce the transcoding latency. In this paper, we propose a new motion vector composition scheme for arbitrarily dropping any frame from incoming video bit-stream comprising I, B and P frames. The transcoded bit-stream retains the I-B-P frame structure. Experimental results are presented and compared to show the efficacy of the proposed scheme.

1. INTRODUCTION

Video transcoding is expected to play an important role for universal multimedia access (UMA) by the Internet users with variety of access links and devices. There is a rapid diversification in the type of network links used to access the Internet, for example, wired or wireless LAN, WAN, DSL, ISDN or cable modem. These network links have different time-varying channel characteristics, *e.g.*, bandwidth, bit error-rate and packet loss-rate. At the same time, more and more portable devices such as hand held computers, personal digital assistants (PDA's) and smart cellular phones have become capable of accessing the Internet. These devices greatly vary in their computing, storage and display capabilities. To achieve the goal of UMA, video contents need to be adapted to various channel conditions and user device capabilities and interests. In this context, transcoding is emerging as a key technology to fulfill the challenge of UMA. Transcoding, by adjusting coding parameters appropriately, can provide much finer and more dynamic adjustment of bit-rate to meet various channel conditions and client requirements [3]. Recognizing this fact, the emerging MPEG-7 standard, has defined ”transcoding hints” to facilitate transcoding of compressed video streams.

When one or more reference frames are dropped from the incoming bit-stream, motion vectors of non-dropped frame become invalid as they may point to the dropped frames which no longer exist in the transcoded bit-stream. Hence it becomes necessary to compose a new set of motion vectors from the current frame to the previous non-dropped frame(s), that will act as a new reference frame(s) in the transcoded bit-stream. The best-matched area, with the current macroblock, overlaps with four macroblocks in the dropped frame. For the composition of the target motion vector bi-linear interpolation of motion vectors of these four macroblocks is added to the current motion vector in [2]. In Forward Dominant Vector Selection (FDVS) method [9], motion vector of the macroblock with largest overlap is selected and then added to the current motion vector to obtain the target motion vector. Some improvements to the FDVS method to reflect the effect of macroblock types in the dropped frames were suggested in [5]. In Activity Dominant Vector Selection (ADVS) method [1], one motion vector is selected from the above four motion vectors based on spatial activity of the macroblock. Telescopic Vector Composition (TVC) [6], accumulates all the motion vectors of the corresponding macroblocks of the dropped frames and add each resultant composed motion vector to its correspondence in the current frame.

However, all the above techniques of motion vector composition are proposed for dropping P-frames from the incoming bit-streams comprising only I and P frames. In order to achieve lower bit-rates for popular I-B-P frame structured video bit-streams any frame, including the I-frame, may be arbitrarily dropped. In many cases, the original frame structure also needs to be retained *e.g.* to facilitate the multi-stage transcoding along the network path. Recently, Bi-directional Dominant Vector Selection method (BDVS) [8] and Bi-directional Telescopic Vector Composition method (BTVC) [4] considered incoming bit-streams comprising I, B and P frames. However, they focused on dropping frames in a fixed sequence and their transcoded bit-stream contained I and P frames only.

In this paper, we propose a new motion vector composition scheme, which we refer, Generic Bi-directional Dominant Vector Selection (GBDVS), for *arbitrarily* dropping any frame from a *generic* video sequence comprising I, B and P frames. The proposed scheme is based on the FDVS method and it retains the I-B-P frame structure in transcoded

0-7803-9332-5/05/$20.00 ©2005 IEEE

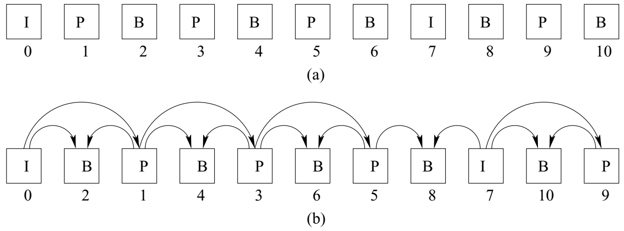

Fig. 1. Typical MPEG-2 video frame pattern: (a) Decoding order, and (b) Display order.

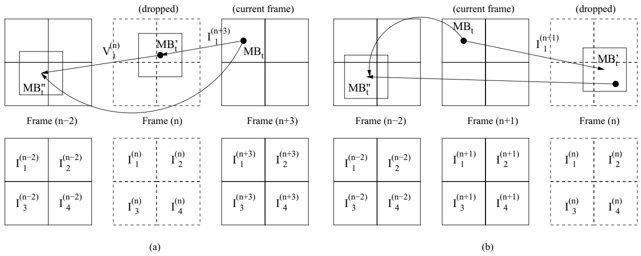
(dropped)
(current frame)
$V_1^{(n)}$
MB_1'
$I_1^{(n+3)}$
MB_1
MB_1''
Frame (n−2)
Frame (n)
Frame (n+3)
$I_1^{(n-2)}$ $I_2^{(n-2)}$ $I_3^{(n-2)}$ $I_4^{(n-2)}$
$I_1^{(n)}$ $I_2^{(n)}$ $I_3^{(n)}$ $I_4^{(n)}$
$I_1^{(n+3)}$ $I_2^{(n+3)}$ $I_3^{(n+3)}$ $I_4^{(n+3)}$
(a)
(current frame)
(dropped)
MB_1
$I_1^{(n+1)}$
MB_1'
MB_1''
Frame (n−2)
Frame (n+1)
Frame (n)
$I_1^{(n-2)}$ $I_2^{(n-2)}$ $I_3^{(n-2)}$ $I_4^{(n-2)}$
$I_1^{(n+1)}$ $I_2^{(n+1)}$ $I_3^{(n+1)}$ $I_4^{(n+1)}$
$I_1^{(n)}$ $I_2^{(n)}$ $I_3^{(n)}$ $I_4^{(n)}$
(b)

Fig. 2. Motion vector composition when P-frame is dropped: (a) Forward frame, and (b) Backward frame.

bit-stream. The experimental results are presented using pixel-domain transcoding architecture. However, the proposed scheme can also be adopted for DCT-domain transcoding architecture.

The rest of the paper is organized as follows. The proposed Generic Bi-directional Dominant Vector Selection (G-BDVS) scheme is presented in section 2. The experimental results are presented and discussed in Section 3. Finally, we conclude the paper in section 4.

2. ARBITRARY FRAME DROPPING

In this section, we describe the proposed motion vector composition algorithm for arbitrary frame dropping in video transcoding. We consider an incoming video sequence - Fig. 1 - for referring to the motion vector composition scheme explained in rest of this section. Consider n^{th} reference frame (I or P frame) in this sequence. Frames (n+1) and (n+3) are, respectively, backward and forward predicted from frame (n). Frame (n+2), if it is P-frame, is also forward predicted from frame (n).

For example, if we take n=1 then frame (2) and frame (4) are B-frames and hence they are, respectively, backward and forward predicted from frame (1). Frame (3) is a P-frame and hence it is also forward predicted from frame (1). For n=5, frame (6) and frame (8) are B-frames and hence they are respectively, backward and forward predicted from frame (5). However, Frame (7), being an I-frame, is independent of frame (5).

When frame (n) is dropped, motion vectors of the frames referring to frame (n) become invalid. We discuss motion vector composition process for frames (n+1) and (n+3). Frame (n+2), if it is P frame, can be handled similar to frame (n+3). As shown in Figs 2 and 3, the forward references in frame (n+3) and backward references in frame (n+1) for macroblocks such as MB_1 become invalid as they point to the dropped frame. In these figures, MB_1' represent best matching block to MB_1 and MB_1'' represents a best matching block to MB_1'.

Case I: When n^{th} Frame is P-Frame: For frame (n+3) in Fig. 2(a), since MB_1' is not on a macroblock boundary, $V_1^{(n)}$ is not available from the incoming bit-stream, hence, a vector addition of $I_1^{(n+3)}$ and $V_1^{(n)}$ to locate MB_1'' is not possible. Three techniques are available in literature to come up with an approximation of $V_1^{(n)}$. One, [2] uses bilinear interpolation of $\{I_1^{(n)}, I_2^{(n)}, I_3^{(n)}, I_4^{(n)}\}$. Two, Activity Dominant Vector Selection (ADVS) [1] uses spatial activity information of the macroblocks, such as the number of non-zero quantized DCT coefficients, to select one of the above four motion vectors. And third, the Forward Dominant Vector Selection (FDVS) [9] method selects a dominant motion vector which is defined as the motion vector carried by a macroblock having largest overlap with MB_1'. We use FDVS method in our work to come up with an approximation of $V_1^{(n)}$. Thus, in Fig. 2(a) $V_1^{(n)} = I_2^{(n)}$. However, above three techniques are proposed to form a target motion vector when incoming bit-stream uses all P-frames and hence can only be used for resolving forward references. In this situation, we propose a following strategy for handling the backward references in frame (n+1). For backward predicted macroblocks of frame (n+1), such as MB_1 in Fig. 2(b), we add $I_4^{(n)}$ to $I_1^{(n+1)}$ to locate MB_1'' in frame (n-2) and then convert the prediction direction of MB_1 from backward to forward. For bi-directional predicted macroblocks we simply convert the prediction direction to forward.

The dominant macroblock for MB_1, found in the above process, may turn out to be intra coded. Since intra coded dominant macroblocks do not carry forward motion vector, we decided MB_1 to be intra coded. Also, we assume zero motion vector for skipped dominant macroblocks. If current macroblock MB_1 is skipped then MB type of the macroblock at same macroblock position in frame (n) is taken as new MB type of MB_1.

Mulitiple reference frame dropping can be handled by cumulatively composing the motion vectors and then storing the motion vectors of the reference frames. We need two tables to store the composed motion vectors corresponding to forward and backward reference frames. For example, if reference frame *i.e.* frame ($n-2$) for frame (n) is also dropped then motion vectors for frame (n) are obtained in

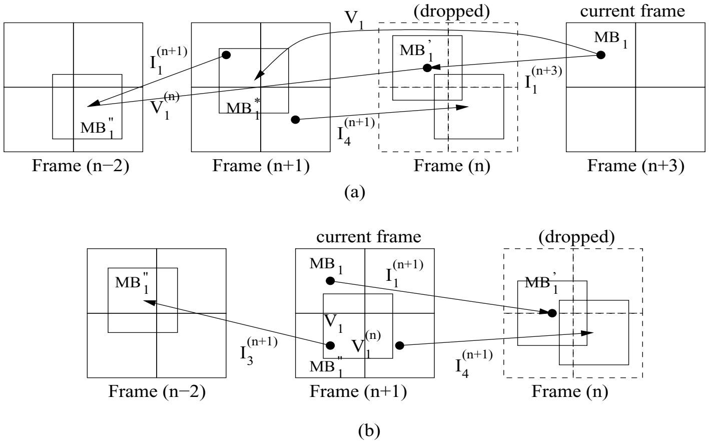

Fig. 3. Motion vector composition when I-frame is dropped: (a) Forward frame, and (b) Backward frame.

similar manner to frame ($n + 2$) above and stored in a forward table. This stored motion vectors in the forward table are then used in the above process to compose motion vectors for frame ($n+1$), frame ($n+2$) and frame ($n+3$). While processing frame ($n + 2$), its composed motion vectors are stored in the backward table. The contents of forward and backward tables are swapped at the occurance of next reference frame in decoding order.

Case II: When n^{th} Frame is I-Frame: For frame (n+3) in Fig. 3(a), since $V_1^{(n)}$ is not available form the incoming stream (as frame (n) is I frame), vector addition of $I_1^{(n+3)}$ and $V_1^{(n)}$ to locate MB_1'' in frame (n-2) is not possible. Rather we derive a vector V_1 locating a best matching block, MB_1^*, in frame (n+1), with MB_1' to continue the process. That is, we obtain a dominant macroblock in frame (n+1) instead of frame(n). Then MB_1'' can be located using $V_1 + I_1^{(n+1)}$ as shown in Fig. 3(a). To derive V_1, we select one motion vector from $\{I_1^{(n+1)}, I_2^{(n+1)}, I_3^{(n+1)}, I_4^{(n+1)}\}$ pointing to a block in frame (n) which has largest overlap with MB_1' and satisfy the dominance criterion (*i.e.*, overlap of 25 or more percent) and then subtract it form $I_1^{(n+3)}$. For the backward references of frame (n+1) in Fig. 3(b), we locate MB_1'' using vector $I_3^{(n+1)} + (I_1^{(n+1)} - I_4^{(n+1)})$ and then convert prediction direction of MB_1 from backward to forward. Again, for bi-directional predicted macroblocks we simply convert prediction direction to forward. It may so happen that none of the motion vector from $\{I_1^{(n+1)}, I_2^{(n+1)}, I_3^{(n+1)}, I_4^{(n+1)}\}$ points to such a block in frame (n). In such cases macroblock MB_1 is intra coded.

There are two major transcoding architectures: the cascaded pixel domain transcoder (CPDT) and the DCT domain transcoder (DDT) – see [7] for review. CPDT decodes the incoming bit-stream in pixel domain and re-encodes it at desired output bit-rate and spatio-temporal resolution. It is more flexible and drift free. In DDT, DCT coefficients are formed by partial decoding and directly processed to achieve desired bit-rate. Although, processing complexity is reduced, DDT lacks the flexibility of CPDT and generally, can not handle spatio-temporal resolution changes without causing considerable drift. We adopt CPDT as our transcoding architecture for frame-skipping transcoder.

3. EXPERIMENTAL RESULTS

The experimental results presented in this section are based on our transcoding implementation using MPEG-2 Test Model 5 (TM5) video codec. To present our results, ”Table Tennis” and ”Football” sequences in SIF (352×240) format are encoded at 4 Mbps and 30 fps with ”IBBPBBPBB...” as the group of pictures (GOP) structure (N=15,M=3). Fig. 4 shows the performance comparison of the proposed GBDVS scheme with Full Search Motion Estimation (FSME), Bi-directional Dominant Vector Selection (BDVS) and Bi-directional Telescopic Vector Selection (BTVC) method, for ”Table Tennis” sequence. Here, the incoming bit-stream is transcoded to 15 fps at 2 Mbps by dropping alternate frames in the display order.

Another result in Fig. 5 shows the performance comparison of the proposed GBDVS scheme with FSME for ”Football” sequence. Here, the incoming bit-stream is transcoded to 15 fps at 2 Mbps by dropping alternate frames in the decoding order. Table 1 shows the average PSNR (dB)and generated bits for the two sequences.

As it can be seen, the proposed GBDVS method outperforms BDVS and BTVC methods both in terms of quality and generated bits. The average PSNR for ”Table Tennis” sequence using proposed GBDVS method is about 0.48 dB and 0.58 dB better than BDVS and BTVC respectively. Also, the average PSNR’s obtained using proposed GBDVS method are close to FSME (0.50 dB for Table Tennis and 0.31 dB for Football). Additional refinement on the com-

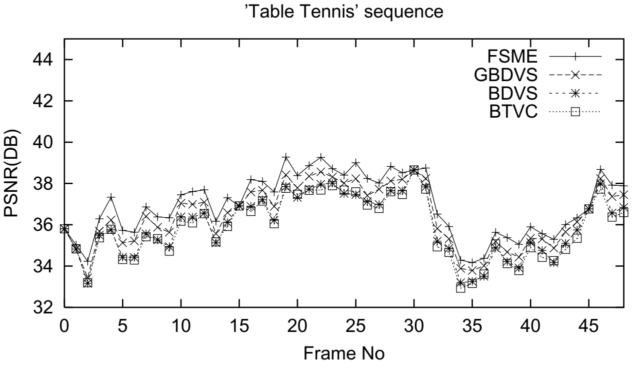

Fig. 4. Performance comparison of proposed GBDVS scheme with different MV composition methods when "Table Tennis" sequence at 30 fps is transcoded to 15 fps by dropping alternate frames in display order.

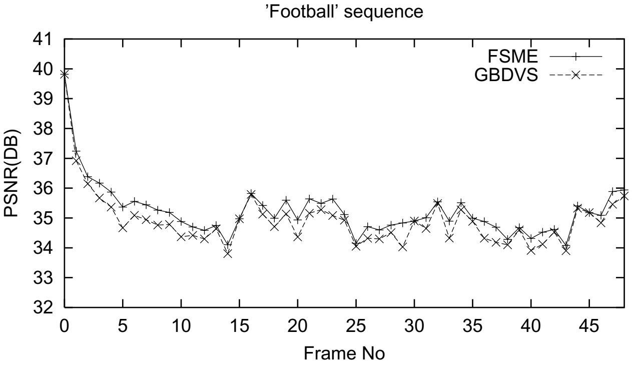

Fig. 5. Performance comparison of proposed GBDVS scheme with FSME when "Football" sequence at 30 fps is transcoded to 15 fps by dropping alternate frames in decoding order.

posed motion vectors can further improve the PSNR obtained by proposed GBDVS to the level of FSME.

4. CONCLUSIONS

In this paper, we have proposed a generic motion vector composition scheme to handle arbitrary dropping of any frame in video bit-streams comprising I, B and P-frame types. An incoming I-B-P frame structure is retained in the transcoded bit-streams. Our experimental results show that the proposed method out-performs existing methods *i.e.* BDVS and BTVC in terms of quality as well as generated bit-rates. The quality achieved and bit-rate generated by the proposed scheme is close to full scale motion estimation.

Table 1. Performance Comparison of MV composition.

Method	*Table Tennis*		*Football*	
	Generated Bits (bytes)	Avg PSNR	Generated Bits (bytes)	Avg PSNR
FSME	1126605	36.96	1304371	35.23
GBDVS	1139265	36.46	1319070	34.92
BDVS	1152697	35.98	-	-
BTVC	1158445	35.88	-	-

5. REFERENCES

[1] M.-J. Chen, M.-C. Chu, and C.-W. Pan, "Efficient motion-estimation algorithm for reduced frame-rate video transcoder," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 4, pp. 269–275, April 2002.

[2] J.-N. Hwang, T.-D. Wu, and C. Lin, "Dynamic frame-skipping in video transcoding," in *Proc. IEEE 2nd Workshop on Multimedia Signal Processing*, pp. 616–621, 1998.

[3] R. Kumar, "A protocol with transcoding to support QoS over internet for multimedia traffic," in *Proc. IEEE International Conference on Multimedia and Expo (ICME'03)*, vol. 1, July 2003, pp. 465–468.

[4] W. J. Lee and W. J. Ho, "Adaptive frame-skipping for video transcoding," in *Proc. IEEE International Conference Image Processing (ICIP'03)*, vol. 1, July 2003, pp. 165–168.

[5] K.-D. Seo, S.-K. Kwon, S. Hong, and J. Kim, "Dynamic bit-rate reduction based on frame-skipping and requantization for MPEG-1 to MPEG-4 transcoder," in *Proc. International Symposium on Circuits and Systems (ISCAS '03)*, vol. 2, May 2003, pp. 372–375.

[6] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding to lower spatial-temporal resolutions and different encoding formats," *IEEE Transactions on Multimedia*, vol. 2, no. 2, June 2000.

[7] A. Vetro, C. Charilaos, and H. Sun, "Video transcoding architectures and techniques: an overview," *IEEE Signal Processing Magazine*, March 2003.

[8] J. Xin, "Improved standard-conforming video transcoding techniques," Ph.D. dissertation, University of Washington, 2002.

[9] J. Youn, M. Sun, and J. Xin, "Video transcoder architectures for bit rate scaling of H.263 bit streams," *ACM Multimedia*, November 1999.