

Motion Re-estimation for MPEG-2 to MPEG-4 Simple Profile Transcoding

Jun Xin, Ming-Ting Sun*, and Kangwook Chun**

*Department of Electrical Engineering, University of Washington

**Samsung Electronics Co. Ltd., Suwon, Kyungki, Korea

Abstract

Since its emerging in mid 90's, MPEG-2 [1] has been widely accepted by the digital video industry. There have been huge amount of video content stored in MPEG-2 format. MPEG-4 [2] is the latest video coding standard out of MPEG targeted at network video applications. The Simple Profile (SP) of MPEG-4 is often used in practical applications. Thus, it is often necessary to perform MPEG-2 to MPEG-4 SP transcoding for the transmission of MPEG-2 coded video content over networks. Such transcoding usually requires hybrid spatial/temporal resolution down-sampling. Since MPEG-4 SP does not support bi-directional prediction, the B-pictures in the MPEG-2 bit-stream need to be changed into P-pictures. In addition, the transcoder needs to be able to deal with interlaced input video supported by MPEG-2 standard. In this paper, we propose techniques to perform motion vector re-estimation that effectively handles the hybrid spatial/temporal resolution reduction with picture type conversion, as well as the interlaced MPEG-2 input video.

I. Introduction

MPEG-2 video is considered to be the most successful digital video coding standard so far. Its applications include digital television broadcasting, digital video disc (DVD), and direct satellite broadcasting, etc. There have been huge amount of MPEG-2 coded video contents. MPEG-4 video is the latest video coding standard from MPEG, which mainly targets at streaming video applications and low bit-rate video applications. MPEG-4 defines several profiles to address different applications. Simple profile (SP) is often used in practical applications. In this paper, we discuss the transcoding from MPEG-2 Main Profile at Main Level (MP@ML) to MPEG-4 Simple Profile at CIF resolution, since MPEG-2 MP@ML is the most widely used MPEG-2 profile/level combination. The discussion in this paper is general to all MPEG-2 profiles and levels.

In [3], the authors investigated techniques for spatial resolution down-sampling, temporal resolution down-scaling and picture type change. However, the techniques were discussed separately, and the authors did not address the strategy that can jointly handle them at the same time. In [4], hybrid spatial/temporal resolution down-scaling was discussed, however no picture type change was considered. Moreover, none of the above works addressed how to handle interlaced input video where frame/field based motion estimation may be used. In this paper we will propose an effective strategy to perform motion vector re-estimation for MPEG-2 MP@ML to MPEG-4 SP transcoding that deals with hybrid spatial/temporal resolution down-scaling with picture type change as well as interlaced input video.

The rest of the paper is organized as follows. Section 2 briefly describes the transcoding structure. The proposed motion re-estimation strategy is explained in Section 3. Section 4 shows simulation results, and section 5 concludes this paper.

II. Transcoding Structure

There are two major transcoder architectures: cascaded pixel domain transcoder (CPDT) [5] and DCT domain transcoder (DDT) [6]. CPDT decodes the input coded video to pixel domain and re-encodes the decoded video into the target format. It is flexible since its decoder-loop and encoder-loop can be totally independent - they can operate at different bit-rates, temporal resolutions, spatial resolutions and even different standards. Also CPDT architecture can be drift-free. DDT directly processes DCT coefficients instead of decoded pixels. However DDT lacks the flexibility of CPDT. Generally, it cannot handle temporal or spatial resolution changes without causing drift. So CPDT is adopted as the transcoder architecture for our MPEG-2 to MPEG-4 transcoding, as shown in Figure 1. The simplified encoder is different from a stand-alone video encoder in that the motion estimation and/or other coding operations may reuse the decoded information from the incoming video stream.

The down-sampling filter changes the input pictures from the CCIR-601 resolution to the CIF resolution. In the format conversion process, only the second field of the decoded video is kept. Then the down-sampling filtering process given in MPEG-4 VM18 [7] is applied to convert the pictures into the CIF resolution.

III. Motion Vector Re-estimation

MPEG-4 Simple Profile is mainly for low bit-rate video applications and usually has lower frame-rate and spatial resolution than those of the MPEG-2 video. The commonly used frame-rates of MPEG-4 SP are 15 fps and 10 fps, while MPEG-2 MP@ML video usually uses 30fps. MPEG-4 SP typically works on CIF and QCIF resolutions, and MPEG-2 MP@ML typically operates at the CCIR-601 [8] resolution.

The temporal resolution down-sampling requires extra handling for motion vectors in addition to spatial handling. To explain the motion re-estimation schemes, we discuss the following typical situations for MPEG-2 MP@ML to MPEG-4 SP transcoding. The input interlaced video of resolution 720x480 is coded using MPEG-2 MP@ML, with frame rate 30fps and GOP structure (15,3). The output is MPEG-4 SP of CIF resolution at 15fps. This frame-rate down-scaling requires picture type change since MPEG-4 Simple Profile does not support B-VOP. We will transcode P and B frames to P-VOPs and I frames to I-VOPs.

As illustrated in Table 1, three different picture-type change patterns exist for such transcoding. The patterns repeat periodically due to the periodic input video GOP structure. The three patterns are classified according to the relative position of the input frame in its GOP, as will be discussed in the following in details. The discussions can be extended to different input GOP structures.

When there are frame-rate reductions, new motion vectors need to be estimated. For example, in Table 1, in the input video, picture $B_{4,in}$ does not have motion vectors using $B_{2,in}$ as

the reference frame, while such motion vectors are necessary for the transcoding ($B_{2,in}$ is input frame number 2, B picture, and $B_{4,in}$ is input frame number 4, B picture.) In this situation, not only the motion information of $B_{4,in}$, but also the motion information of $B_{2,in}$ will be used to derive the motion vectors for $P_{2,out}$.

For the ease of explanation, we break this motion vector re-estimation process into two steps: 1) temporal resolution down-scaling and 2) spatial resolution down-scaling.

In the following discussions Table 1 is used unless otherwise indicated.

Step 1: Candidate motion vectors with temporal resolution down-scaling

In this step, we will explain the general principle to find the motion vectors of a macroblock at the same resolution as the input picture for the three different patterns of picture type changes.

Picture type change – pattern 1

The conversion of $B_{2,in}$ to $P_{1,out}$ is of pattern 1. The target motion vectors here are for $B_{2,in}$ to $I_{0,in}$. Some macroblocks of $B_{2,in}$ may have forward motion vectors using $I_{0,in}$ as the reference, while others may not. For either case, target motion vectors can be obtained using the following strategies:

- $MV_{2 \rightarrow 0}$: Forward MV from $B_{2,in}$ to $I_{0,in}$.
- $MV_{3 \rightarrow 0} + MV_{2 \rightarrow 3}$: Forward MV for $P_{3,in}$ to $I_{0,in}$ plus the backward MV from $B_{2,in}$ to $P_{3,in}$.

Picture type change – pattern 2

In this case, $B_{4,in}$ is converted to $P_{2,out}$. Thus the target motion vectors are from $B_{4,in}$ to $B_{2,in}$. These motion vectors can be obtained using this way:

- $MV_{4 \rightarrow 3} - MV_{2 \rightarrow 3}$: Forward MV from $B_{4,in}$ to $P_{3,in}$ minus backward MV from $B_{2,in}$ to $P_{3,in}$.

$B_{4,in}$ and $B_{2,in}$ are just one frame away from frame $P_{3,in}$ in time, so for most macroblocks $MV_{4 \rightarrow 3}$ and $MV_{2 \rightarrow 3}$ exist and the target motion vector can be computed. In case one of these two motion vectors does not exist, the target motion vector can be obtained as follows:

$$MV_{4 \rightarrow 3} = MV_{6 \rightarrow 3} + MV_{4 \rightarrow 6}$$

$$MV_{2 \rightarrow 3} = MV_{2 \rightarrow 0} - MV_{3 \rightarrow 0}$$

Picture type change – pattern 3

In this case, $P_{6,in}$ is converted to $P_{3,out}$. Although the picture type is not changing, the reference picture is different. The target motion vectors are from $P_{6,in}$ to $B_{4,in}$. Motion vectors are formed using the following methods:

- $-(MV_{4 \rightarrow 6})$: reverse the backward MV from $B_{4,in}$ to $P_{6,in}$.
- $MV_{6 \rightarrow 3} - MV_{4 \rightarrow 3}$: forward MV from $P_{6,in}$ to $P_{3,in}$ minus forward MV from $B_{4,in}$ to $P_{3,in}$.

More details of finding the motion vectors

To support interlaced video, for frame-pictures, MPEG-2 specifies that each of its macroblock could have either two field motion vectors or one frame motion vector. Recall that the spatial resolution down-sampling process involves discarding the first field of the input frame. Suppose the top field comes first, then only the bottom field is retained to form the target picture. So the motion vectors we are trying to form in this step are for the bottom field and use the

bottom reference field as the reference field. For some macroblocks, there may exist such motion vectors. These motion vectors will be used in the motion vector composition in the second step, and are called “normal” motion vectors [9]. For other macroblocks, there may not be such motion vector. For example, a macroblock may have a motion vector for the bottom field that refers to the top reference field, but not to the bottom reference field. For these macroblocks we can derive the desired motion vector by properly scaling the existing motion vector. Figure 2 illustrates how to derive a motion vector of the bottom field referring to the bottom reference field from the motion vector referring to the top reference field. Similarly, we can derive a motion vector for the bottom field referring to the bottom reference field from a frame motion vector. Motion vectors derived in this way, along with those normal motion vectors, are called “extended” motion vectors, and will both be used in the motion vector composition in the next step [9].

Using the procedures described above, multiple motion vectors for a macroblock usually are formed. These formed motion vectors are just stored for use in the second step and no motion estimation is performed.

It is likely that no motion vector is found for a macroblock through the above processes. For those situations, that macroblock is labeled as “NO-MV” macroblock, meaning that no motion vectors of that macroblock will be used in the second step of motion vector composition.

Step 2: Motion vector composition for spatial resolution down-scaling

With the extended candidate motion vectors obtained in the first step, what remains is to find motion vectors for each macroblock of the spatially down-sampled target video frames.

Figure 3 illustrates the mapping area (in shadow) in the input picture from which the target macroblock is down-sampled. The vertical down-sampling ratio is $480:288=5:3$, and this figure shows that 6 macroblocks are overlapping with the mapping area of target macroblock. Clearly motion vectors of these macroblocks correlate with the motion of the target macroblock, and are chosen to be the candidate motion vectors. Each candidate motion vector characterizes the motion of part of the target macroblock, and the majority of them should be a reasonable measure of the motion of the target macroblock. We propose to use the weighted median of the extended candidate motion vectors to compose the target motion vector. The weight for each motion vector is the overlapping area between the supporting area of the candidate motion vector and the mapping area of the target macroblock. The weighted median operation is expressed in the following expression:

$$\begin{cases} mv' = \frac{1}{R} mv^m, & mv^m \in \{mv_i\} \\ \sum_{i=1}^N w_i \|mv^m - mv_i\| \leq \sum_{i=1}^N w_i \|mv_j - mv_i\| & j = 1, 2, \dots, N \end{cases}$$

where mv' is the final composed motion vector, $\{mv_i\}$ are the extended motion vectors formed as explained in the first section, w_i is the weight associated with each mv_i .

Refinement then is conducted at positions surrounding the composed motion vector. As will shown, half pixel refinement is usually enough to achieve satisfactory performance.

IV. Simulation Results

In the simulations, 60 frames of “Flower” and “Football” sequence which have a resolution of $720 \times 480i$ and 4:2:0 chrominance sampling are used. Both have a frame rate of 30fps, and are encoded using MPEG-2 frame pictures at 5 Mb/s. They are transcoded to MPEG-4 SP CIF resolution of 15fps. The target bit-rate is 768 kbps. TM-5 rate-control is used. We simulated the full-search with search-range (-8,+8) to evaluate the performance of the proposed algorithm. Table 2 shows the simulation results. Figure 4 shows the PSNR comparison between the proposed algorithm and the full search. It can be seen that the performance of the proposed algorithm approaches the full-search algorithm without doing the computationally expensive full-search. Through our computer simulation, when only the computation of the motion estimation is compared, the proposed approach achieves saving of more than 80%. In the “Football” sequence, it actually outperforms the full-search algorithm. Part of the reason may be that the proposed weighted median approach produces smoother block motion vector field than the full search algorithm, which may save bits in coding the differential motion vectors. Another possible reason is that the search range of the full search (-8,+8) is not large enough for some macroblocks, while the proposed approach is not restricted by this range. It can be expected the full search approach can improve its performance by increasing the search range, however that will require more computational power.

V. Conclusions

An MPEG-2 MP@ML to MPEG-4 SP transcoder is presented in this paper. In the transcoder, we propose a two-step motion re-estimation strategy to handle the hybrid spatial/temporal resolution down-sampling with picture type changes. We propose approaches to handle the different MPEG-2 macroblock coding modes (frame/field) that supports interlaced input video. Simulations show that the proposed motion re-estimation approach can achieve performance comparable to full-search with much less computations.

References

- [1] ISO/IEC 13818-2, “General coding of moving pictures and associated audio information: Video”.
- [2] ISO/IEC 14496-2, “Coding of audio-visual objects: Visual”.
- [3] T. Shanableh, and M. Ghanbari, “Heterogeneous Video Transcoding to Lower Spatio-Temporal Resolutions and Different Encoding Formats”, IEEE Trans. Multimedia, Vol.2, No.2, June 2000.
- [4] G. Shen, B. Zeng, Y.-Q. Zhang, and Ming L. Liou, “Transcoder with arbitrarily resizing capability”, ISCAS’2001.
- [5] Huifang Sun, Wilson Kwok, and Joel W. Zdepski, “Architectures for MPEG Compressed Bitstream Scaling”, IEEE Trans. Circuits and Systems for Video Technology, Vol. 6, No. 2, April 1996.

- [6] P.A.A. Assuncao and M. Ghanbari, "A frequency-domain video transcoder for dynamic bitrate reduction of MPEG-2 bit streams", IEEE Trans. Circuits and Systems for Video Technology, Vol. 8, pp. 953-967, Dec. 1998.
- [7] ISO/IEC JTC1/SC29/WG11, "MPEG-4 Video Verification Model version 18.0", Section 2.2.2, January 2001, Pisa, Italy
- [8] CCIR Recommendation 601.
- [9] Susie J. Wee, John G. Apostolopoulos, and Nick Feamster, "Field-to-frame Transcoding with Spatial and Temporal Downsampling", IEEE Intl. Conf. Image Processing, 1999, Vol. 4, pp. 271-275.

Table 1: Picture type conversions in MPEG-2 to MPEG-4 SP transcoding. The first two rows are the input frame numbers and input frame coding types. The next two rows are the target (output) frame coding types and numbers. The last row gives the conversion pattern number of picture type conversion which is explained in the text.

Input number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Input	I	B	B	P	B	B	P	B	B	P	B	B	P	B	B	I	B
Output	I		P		P		P		P		P		P		P		P
Output number	0		1		2		3		4		5		6		7		8
Conversion Pattern			1		2		3		1		2		3		1		2

Table 2: Performance (average PSNR, in dB) of proposed algorithm (weighted median of extended candidates with $\frac{1}{2}$ pixel refinement) vs. full search.

Sequence	Full Search (± 8)	Proposed
Flower	28.66	28.23
Football	32.26	32.34

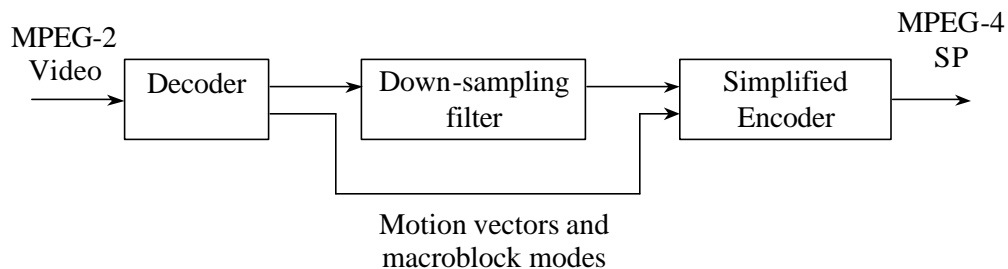


Figure 1: CPDT transcoder structure.

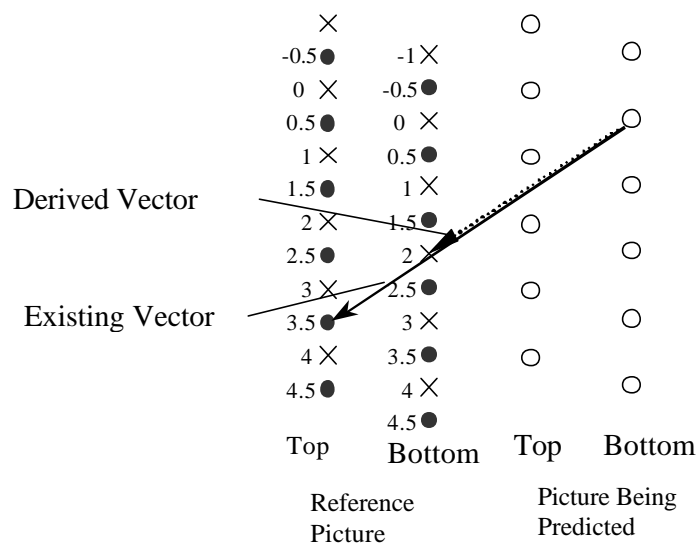


Figure 2: Derive a field motion vector for a different reference field. In this illustration, a motion vector of the current bottom field using the bottom reference field is derived from a motion vector using the top reference field.

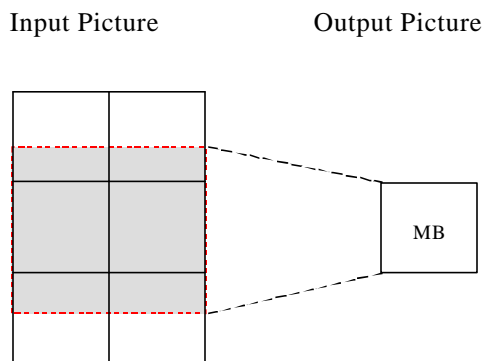


Figure 3: Mapping area, candidate macroblocks of a target macroblock (MB).

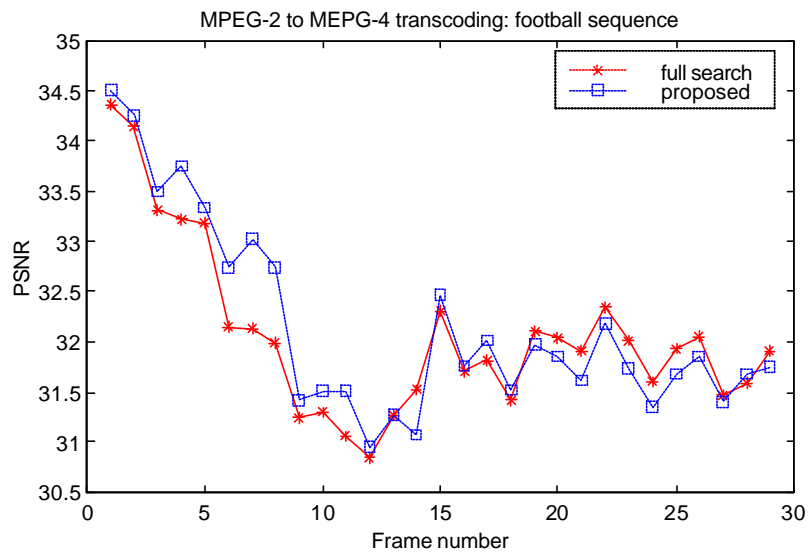


Figure 4: Performance comparison: full search vs. proposed motion re-estimation.