Efficient MPEG-4 to H.264/AVC Transcoding with Spatial Downscaling

Toan Dinh Nguyen, Guee-Sang Lee, June-Young Chang, and Han-Jin Cho

ABSTRACT—Efficient downscaling in a transcoder is important when the output should be converted to a lower resolution video. In this letter, we suggest an efficient algorithm for transcoding from MPEG-4 SP (with simple profile) to H.264/AVC with spatial downscaling. First, target image blocks are classified into monotonous, complex, and very complex regions for fast mode decision. Second, adaptive search ranges are applied to these image classes for fast motion estimation in an H.264/AVC encoder with predicted motion vectors. Simulation results show that our transcoder considerably reduces transcoding time while video quality is kept almost optimal.

Keywords—MPEG-4 part 2, H.264/AVC, transcoder, motion estimation, mode decision, spatial downscale.

I. Introduction

MPEG-4 has been widely used in many applications, such as cell phones, video conferencing systems, and digital multimedia broadcasting (DMB) [1]. Recently, H.264/AVC has been presented as an up-to-date video coding standard with good video quality at a substantially lower bit rate than MPEG-4 part 2. Since there are many applications which employ the H.264/AVC standard, there is a strong need for an efficient MPEG-4 (with simple profile) to H.264/AVC transcoder [2].

Transcoding can be performed in the spatial or transform domain [2] and with or without the reduction of temporal [3] or spatial [4] resolution to meet the constrained transmission bandwidth and terminal capabilities. Although there has been extensive research on both homogeneous and heterogeneous video transcoding, MPEG-4 to H.264/AVC transcoding has begun rather recently [5], [6]. A few algorithms for MPEG-4 to H.264/AVC transcoding have been recently presented [5], [6], but spatial downscaling has not been dealt with in the literature. In [4], a transcoding method based on merge and partition was applied for MPEG-2 to H.264/AVC transcoding with spatial resolution reduction. In MPEG-4, the motion estimation module considers 4 modes (inter 16×16, inter 8×8, SKIP, and intra) rather than 1 mode as in MPEG-2 (inter 16×16). Therefore, a simple application of the method proposed in [4] to MPEG-4 to H.264/AVC transcoding incurs too many cases to handle.

In this letter, we present an efficient method for MPEG-4 to H.264/AVC transcoding with spatial downscaling. The image characteristic of the target block is classified into one of three groups, monotonous, complex, or very complex regions, for fast mode decision. The proposed algorithm adopts adaptive search ranges for different regions, classified by image complexity, for fast motion estimation. In existing MPEG-4 to H.264/AVC transcoders [5], [6] reduced search ranges are not properly exploited.

II. Architecture and Spatial Downsampling

Two major transcoding architectures are the cascaded pixel domain transcoder (CPDT) and the DCT domain transcoder (DDT) [1]. The DDT directly processes DCT coefficients, and it is difficult for it to handle spatial resolution reduction without causing drift. The CPDT is flexible and drift-free because its decoder-loop and encoder-loop are independent. We chose CPDT for our transcoding architecture.

Our transcoder (see Fig. 1) mainly focuses on spatial downsampling and the motion information reusing module. In the

Manuscript received May 3, 2007; revised Oct. 1, 2007.

This work was supported by the IT R&D program of MIC/IITA, Rep. of Korea [2005-S077-02, Development of On-Chip Network Based SoC Platform].

Toan Dinh Nguyen (email: toan_mulmi@hotmail.com) and Guee-Sang Lee (phone: + 82 62 530 3425, email: gslee@jnu.ac.kr) are with the Department of Electronics and Computer Engineering, Chonnam National University, Gwangju, Rep. of Korea.

June-Young Chang (email: jychang@etri.re.kr) and Han-Jin Cho (email: hjcho@etri.re.kr) are with the IT Convergence & Components Laboratory, ETRI, Daejeon, Rep. of Korea.

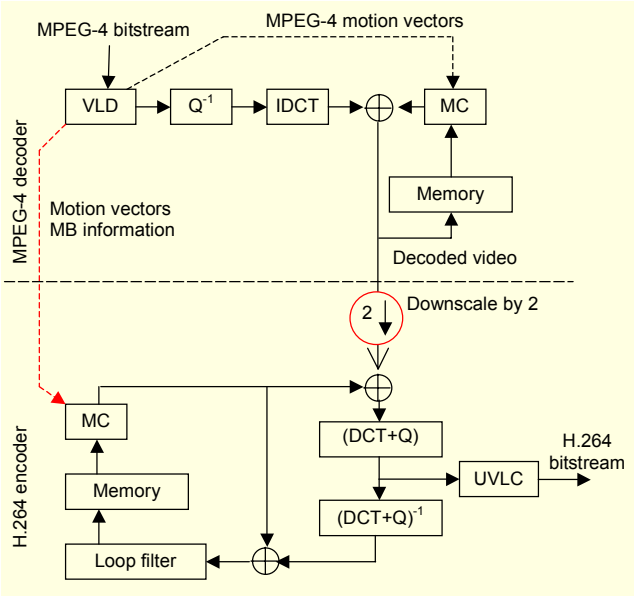

Fig. 1. CPDT transcoder architecture with reuse of motion information and spatial resolution reduction by 2.

architecture, the input bitstream is decoded and encoded frame by frame with the MPEG-4 decoder and the H.264/AVC encoder, respectively. The decoded frame data is spatially downscaled 2:1 in each direction, that is, 4 macroblocks (MBs) into 1 MB, before being encoded again by the H.264/AVC encoder.

III. Motion Estimation with Adaptive Search

1. Fast Mode Decision Method

In the transcoder with 2:1 spatial resolution reduction, the coding mode for every MB in the H.264/AVC encoder is generated from coding modes of four decoded MBs at the same position in MPEG-4. The coding mode with the smallest rate distortion, as defined in the H.264/AVC standard, is chosen among modes in each of four groups. Note that by dividing the group of modes to be considered by image complexity, the computation time can be reduced considerably.

1) For I-frame: Re-encode as intra frame in H.264/AVC encoder.

2) For P-frame:

• If modes of all four decoded MBs are SKIP or inter 16×16 mode, the content of the image region is assumed to be monotonous or low motion, so the H.264/AVC encoder considers only two modes (SKIP and inter 16×16).

• If the modes of the four decoded MBs are with large modes (SKIP or inter 16×16) and small (inter 8×8) or intra modes together, the image region is assumed to be complex or large motion, so the downscaled MB is encoded by small block modes (16×8, 8×16, 8×8, 8×4, 4×8, 4×4).

• If all decoded MBs have only inter 8x8 or intra modes, the region is very complex, so six modes (8×8, 8×4, 4×8, 4×4, intra 16, intra 4) are considered.

2. Reuse of PMVs and Motion Vector

The PMV_{reused} (predicted motion vector for motion reuse) is calculated to find the starting point for the motion search of the downscaled MB.

Step 1. Generate a motion vector for every pre-encoded MB which is downscaled from the 4 adjacent MBs. If a pre-encoded MB has SKIP or intra mode, the downscaled MV is 0. Otherwise, compute the median of 4 motion vectors in a decoded MB. Then, divide by two.

Step 2. After downscaling MV values, set the PMV_{reused} for the H.264/AVC encoder. With large block modes, the image region is considered to be flat, and the median value of previous vectors is computed. On the other hand, if the small block modes are considered, PMV_{reused} is set by the MV of the corresponding sub-macroblock (see Fig. 2).

3. Adaptive Integer Search Range

We can conjecture that complex images with large motion search need a large search range; therefore, we set the search range according to the complexity of the image region to speed up encoding while maintaining an acceptable image quality. In our transcoder, we set the integer search range to 1 for monotonous, 4 for complex, and 8 for very complex regions. According to the analysis of discrepancy between optimal motion vectors and PMVs, the proposed search range can guarantee enough space in most cases.

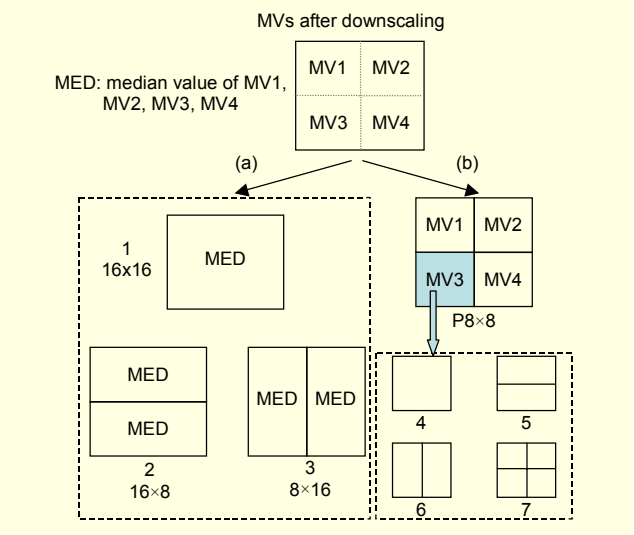

Fig. 2. Calculating PMV_{reused}: (a) large block modes, PMV_{reused} = median value and (b) small block modes, PMV_{reused} = downscaled value at same position.

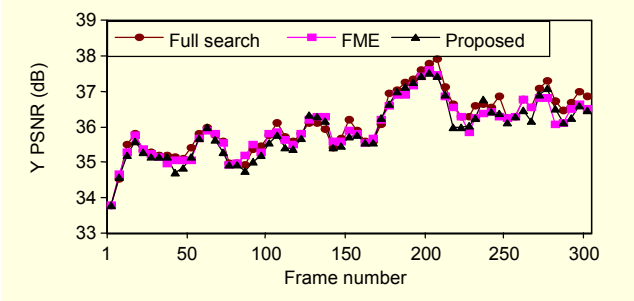

Fig. 3. PSNR for Akiyo sequence.

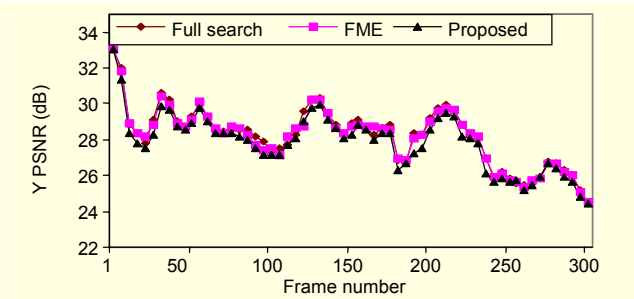

Fig. 4. PSNR for Stefan sequence.

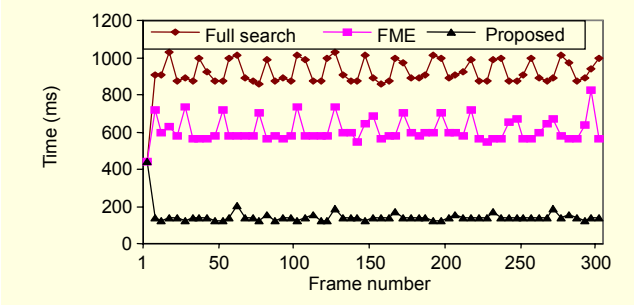

Fig. 5. Total transcoding time for Akiyo sequence.

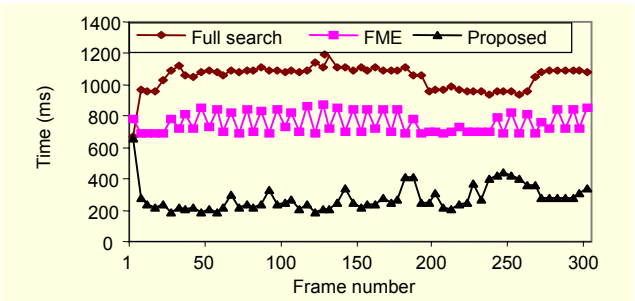

Fig. 6. Total transcoding time for Stefan sequence.

IV. Experimental Results

We used two CIF video sequences "Akiyo" and "Stefan" to evaluate the performance of our transcoder. Both sequences are encoded by the MPEG-4 encoder using GOP structure IPPP at 30 Hz. The Akiyo sequence was encoded with 112 kbps and transcoded to 20 kbps. The Stefan sequence was encoded with 672 kbps and transcoded to 168 kbps, and 300 frames were considered in each sequence. For motion search, full search and fast motion estimation (FME) [7], which uses a hexagon type search, and our proposed method have been considered. For the full search and FME schemes, we used a search range of 16. The simulation was run on a 2.6 GHz Pentium IV computer with 1 GB RAM. All transcoders were implemented based on the *MPEG4@SP MomuSys-FPDAM1-1.0-020414_nctu* decoder and the *H.264@ MP JM (Joint Model) Ver:11.0* encoder. As shown in Figs. 3 and 4, our method maintained good image quality with only 0.12 dB loss compared to the full search method and 0.07 dB less than that of FME. As shown in Fig. 5, with the Akiyo sequence, which has low motion with many large modes in decoded frames, our transcoding algorithm is about 4 times faster than the FME method. In a high motion sequence, Stefan (see Fig. 6), our method is more than three times faster than the FME method.

V. Conclusion

In this letter, an efficient transcoder with spatial downscaling was presented. Our method significantly reduced transcoding time while preserving almost optimal image quality by using image characteristics for fast mode decision and adaptive search range for motion estimation.

References

[1] B.H. Lee, K.T. Yang, Y.K. Hahm, S.I. Lee, and C.T. Ahn, "A Framework for MPEG-4 Contents Delivery over DMB," *ETRI Journal*, vol. 26, no. 2, Apr. 2004, pp. 112-121.

[2] A. Vetro, C. Christopoulos, and H. Sun, "Video Transcoding Architectures and Techniques: An Overview," *IEEE Signal Processing Magazine*, vol. 20, Mar. 2003, pp. 18-29.

[3] Y.K. Lee, S.S. Lee, and Y.L. Lee, "MPEG-4 to H.264 Transcoding with Frame Rate Reduction," *Multimedia Tools and Applications Journal* (On-line publication), Springer, Netherlands, 2007.

[4] B. Hu, P. Zhang, Q. Huang, and W. Gao, "Reducing Spatial Resolution for MPEG-2 to H.264/AVC Transcoding," *Proc. Pacific-Rim Conf. Multimedia*, part II, LNCS 3768, vol. 2, 2005, pp. 830-840.

[5] S.E. Kim, J.K. Han, and J.G. Kim, "Efficient Motion Estimation Algorithm for MPEG-4 to H.264 Transcoder," *IEEE Int'l Conf. Image Processing*, vol. 3, Sept. 2005, pp. 656-659.

[6] Y. Liang, X.i Wei, I. Ahmad, and V. Swaminahan, "MPEG-4 to H.264/AVC Transcoding," *Int'l Conf. Comm. and Mobile Computing*, 2007, pp. 689-693.

[7] JVT Reference Software Version JM11.0: http://bs.hhi.de/~suehring/tm1/.