A FAST AND MEMORY EFFICIENT VIDEO TRANSCODER FOR LOW BIT RATE WIRELESS COMMUNICATIONS

Shizhong Liu and Alan C. Bovik

Laboratory for Image and Video Engineering, Dept. of Electrical and Computer Engineering
The University of Texas at Austin, Austin, TX 78712-1084, USA
Email: {sliu2, bovik}@ece.utexas.edu

ABSTRACT

Wireless video is one of the important applications supported by upcoming 3G mobile communication systems. In this paper, we propose a fast and memory efficient DCT-domain video transcoder to convert a high quality MPEG2 video bit stream into a low bit rate MPEG4 stream with low spatial resolution for wireless video access. Compared to existing approaches, the proposed video transcoder can save more than 50% of required memory. Furthermore, the computational complexity of the proposed method is less than 30% of that required by existing methods. However, the video quality achieved by the proposed method and by existing methods is hardly distinguishable for target bit rates of 384 kb/s and 256 kb/s, as shown in our experimental results.

1. INTRODUCTION

With the advent of wireless communication systems and client devices such as personal digital assistants (PDAs), hand-held computers and smart phones, it is believed that wireless video communications will be ubiquitous in the near future. Compared to wireline video systems, wireless video communication systems have the following limitations:

- Lower bandwidth.
- Higher bit error rate due to fading effects.
- Limited capability of terminal devices. Most wireless devices have small display screens, limited processing power and memory size.

MPEG4 is the primary video coding standard used in wireless video communications due to its high compression efficiency and strong error resilience. Most mobile devices are or will be designed to support MPEG4 compatible video decoding. However, video content available from the Internet is usually encoded in other formats for different purposes. For example, in video on demand (VOD) systems, the video contents stored in the video server are usually encoded in MPEG2 format with high visual quality and spatial resolution, resulting in bit rates from 4 Mb/s up to 15 Mb/s. In order to enable wireless access to the video stored in the VOD server, video transcoding techniques can be employed to conduct both video coding format and bit rate conversions.

In this paper, we focus on transcoding a high quality MPEG2 video to a low bit rate MPEG4 video with reduced spatial resolution for wireless video access. To do that, we choose to process the incoming MPEG2 video as follows:

1. Reduce the spatial resolution by 2 in both horizontal and vertical directions.
2. Re-encoding all B frames and I frames (except the first I frame) in the MPEG2 video as P frames in the outgoing MPEG4 video.

A straightforward way to scale down the compressed video sequence is to fully decompress the video bit stream, then down sample the video and finally re-encode the reduced video by the MPEG4 encoder. However, the complexity of this approach is very high because it includes both an MPEG2 decoder and an MPEG4 encoder. Instead of simply cascading the video decoder and encoder, various efficient video transcoding techniques have been developed by taking advantage of the information extracted from the incoming video bit stream (*e.g.*, motion vectors, macro-block coding type as well as bit allocation statistics), such that the complexity of video encoding can be reduced significantly [1, 2, 3, 4]. For instance, the motion vectors extracted from the input video can be reused [2] or employed to derive new motion vectors in the video encoder [4, 5], such that a full scale motion estimation, which comprises more than 60% of the encoding complexity, can be avoided. According to [4], the video transcoder can be implemented as illustrated in Fig. 1. It first decodes the incoming MPEG2 bit stream to the pixel-domain by performing variable length

This work was supported in part by Texas Instruments, Inc. and by the Texas Advanced Technology Program.

decoding (VLD), inverse quantization (IQ), inverse DCT (IDCT) and motion compensation (MC); then down-scales the decoded video by half in the pixel-domain; and finally re-encodes the down-scaled video into an outgoing MPEG4 video bit stream. Note that the MC is performed using the original motion vectors. Techniques for estimating the new motion vectors, for the reduced outgoing video, from the extracted motion vectors have also been discussed in [4].

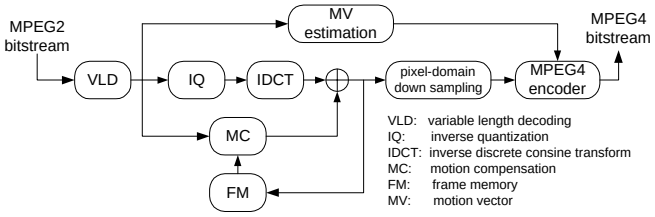

Fig. 1. Pixel-domain video transcoder.

In this paper, we propose a fast and memory efficient DCT-domain video transcoder illustrated in Fig. 2, where the input video is directly decoded to a low resolution video by a so-called DCT-domain down-scale video decoder. Compared to previous approaches, the proposed video transcoder can save more than 50% of required memory. Furthermore, the computational complexity of the proposed method is less than 30% of that required by pixel-domain methods. However, the video quality achieved by both methods is hardly distinguishable at target bit rates of 384 kb/s and 256 kb/s, as shown in our experimental results.

In Section 2, the DCT-domain down-scaled decoder is described in detail. Section 3 discusses how to estimate the new motion vectors for the down-scaled video sequence. Experimental results are given in Section 4 and Section 5 concludes this paper.

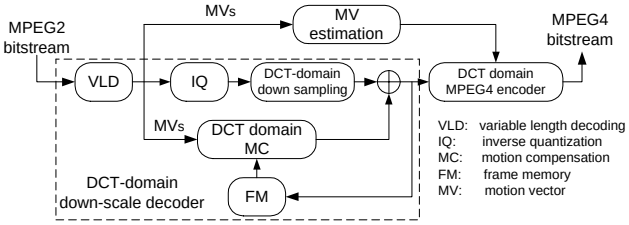

Fig. 2. DCT-domain video transcoder.

2. DCT-DOMAIN DOWN-SCALE DECODER

2.1. DCT-domain down sampling

In each incoming 8×8 block, only top-left 4×4 DCT coefficients are parsed during variable length decoding as shown in Fig. 3. Then, every four 4×4 DCT blocks are

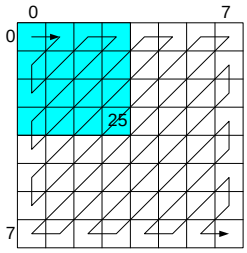

Fig. 3. Variable length decode the top-left 4×4 sub-block in each 8×8 DCT block.

transformed into one 8×8 DCT block in the DCT-domain. To show this, let T and T_4 denote the 8×8 and 4×4 DCT operator matrices, respectively. Then we have

$$\begin{aligned} B &= TbT^t \\ &= \begin{bmatrix} T_L & T_R \end{bmatrix} \begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \end{bmatrix} \begin{bmatrix} T_L^t \\ T_R^t \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} T_L & T_R \end{bmatrix} \begin{bmatrix} T_4^t B_1 T_4 & T_4^t B_2 T_4 \\ T_4^t B_3 T_4 & T_4^t B_4 T_4 \end{bmatrix} \begin{bmatrix} T_L^t \\ T_R^t \end{bmatrix} \\ &= \frac{1}{2}(T_L T_4^t)[B_1 (T_L T_4^t)^t + B_2 (T_R T_4^t)^t] \\ &\quad + \frac{1}{2}(T_R T_4^t)[B_3 (T_L T_4^t)^t + B_4 (T_R T_4^t)^t] \end{aligned} \quad (1)$$

where b, b_1, b_2, b_3, b_4 are the pixel-domain representations of the DCT blocks B, B_1, B_2, B_3, B_4 (shown in Fig. 4), respectively; T_L, T_R are 8×4 matrices denoting the first and last four columns of the eight-point DCT kernel T, respectively; and t denotes matrix transposition. Let $C = T_L T_4^t + T_R T_4^t$ and $D = T_L T_4^t - T_R T_4^t$, then we have $T_L T_4^t = \frac{C+D}{2}$ and $T_R T_4^t = \frac{C-D}{2}$. Hence, (1) can be rewritten as

$$\begin{aligned} B &= \frac{1}{8}\{[C(B_1 + B_3) + D(B_1 - B_3)](C + D)^t \\ &\quad + [C(B_2 + B_4) + D(B_2 - B_4)](C - D)^t\} \\ &= \frac{1}{8}[X(C + D)^t + Y(C - D)^t] \\ &= \frac{1}{8}[(X + Y)C^t + (X - Y)D^t] \end{aligned} \quad (2)$$

with

$$X = C(B_1 + B_3) + D(B_1 - B_3) \quad (3)$$

$$Y = C(B_2 + B_4) + D(B_2 - B_4). \quad (4)$$

It can be shown that more than 50% of the elements in the matrices C and D are zeros, such that (2)-(4) can be computed very efficiently [6].

2.2. Low resolution motion compensation

After down-sampling each frame, we have to convert all B and P frames in the original video sequence back to I

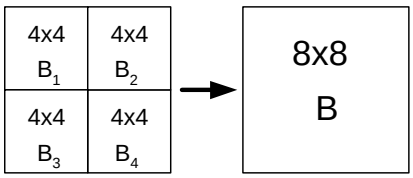

Fig. 4. Convert four 4×4 DCT sub-blocks to one 8×8 DCT block.

frames for the following MPEG4 encoder. Due to the down-sampling, each macro-block in the original video collapses to one block for luminance component (here we assume the original video format is 4:2:0). Accordingly, the motion vectors extracted from the incoming video sequences should also be down-scaled by half in both horizontal and vertical directions. Therefore, the motion vector with half-pixel resolution in the original video sequence has quarter-pixel precision in the reduced video. The DCT-domain motion compensation method proposed in [7] is employed to convert all inter-coded blocks to intra-coded blocks. For chrominance components, we choose to do motion compensation first and then conduct DCT-domain down-sampling.

2.3. Complexity

In the pixel-domain transcoder shown in Fig. 1, each DCT block in the incoming MPEG2 video bit stream undergoes IDCT, down-sampling (replacing each 2×2 block with its average value), and DCT in the MPEG4 encoder (we assume the short header coding mode in MPEG4 standard is used). It can be shown that even if a fast algorithm for DCT and IDCT is employed, the computational complexity of this process would be 3.44 multiplications and 9.81 additions per pixel of the original video [6]. However, it only needs 1.25 multiplications and 1.25 additions per pixel of the original image to obtain the down-scaled DCT block from four DCT blocks of the original image by computing (2)-(4) [6]. Therefore, the computational complexity of the proposed method is less than 30% of that required in the pixel-domain approach shown in Fig. 1. Furthermore, in the pixel-domain approach, MC in the decoder is performed in the full resolution while it is performed in the reduced resolution in the DCT-domain down-scale decoder. By using the DCT-domain motion compensation algorithm proposed in [7], MC in the down-scaled decoder can be implemented faster than that in pixel-domain approaches.

3. MOTION VECTOR ESTIMATION

To avoid full scale motion estimation in the MPEG4 encoder, the motion vectors extracted from the MPEG2 bit stream can be employed to estimate the new motion vectors for the reduced video sequence. Due to the down-sampling, one macro-block in the down-scaled video corresponds to four macro-blocks in the original video sequence. Each inter-coded macro-block in the original video has one motion vector. Various methods have been proposed to derive a new motion vector for the down-scaled macro-block from the corresponding four motion vectors in the original video sequence [4, 5]. In these methods, intra-coded or skipped macro-blocks are usually viewed as predicted macro-blocks with zero valued motion vector. In [5], the authors estimated the new motion vector by using the weighted average of the four incoming motion vectors, where the weights correspond to the block activity. Since in our down-scaled video decoder, the original DCT blocks are not fully decoded, this method is not suitable for our application. Shanableh *et al.* [4] investigated three different methods to derive the new motion vector, *i.e.*, median value, moving with majority and the mean value of the four input motion vectors. They showed that the median value method gives the best results. The median vector is defined as one of the four motion vectors that has the least Euclidean distance from all, *i.e.*

$$\bar{v} = arg \min_{v_k \in V} \{\sum_{\substack{j=1 \\ j \neq k}}^{4} ||v_k - v_j||\} \tag{5}$$

where $V = \{v_1, v_2, v_3, v_4\}$ and $v_j, j = 1, \ldots, 4$ are the incoming motion vectors and $\bar{v}$ is the candidate motion vector for the down-scaled video. Note that the magnitude of the estimated vector v should be scaled down by half. This method is adopted in our experiments. Based on the estimated motion vector, a ± 0.5 pixel motion vector refinement is conducted in the MPEG4 encoder, which is sufficient to obtain the optimal motion vector according to [4].

4. EXPERIMENTAL RESULTS

The video sequence *mobile*, with a spatial resolution of 704×480, is encoded into an MPEG2 bit stream at the bit rate of 6 Mb/s. The MPEG2 bit stream is then transcoded to a MPEG4 video bit stream with half resolution, *i.e.* 352×240, by both the pixel-domain transcoder and the one proposed in this paper, respectively. PSNR values are employed to evaluate the video quality of the MPEG4 bit streams generated by both video transcoders. Since the original low resolution video sequence *mobile* is unavailable, the down-sampled version of the original video sequence *mobile* is used as the reference for PSNR value computations.

In the experiments, the MPEG2 bit stream is transcoded into an MPEG4 bit stream at the target bit rates of 384 kb/s and 256 kb/s, respectively. The PSNR values of the output MPEG4 video sequences are plotted in Fig. 5 and Fig. 6, respectively. Compared to the PSNR values achieved by the pixel-domain method, the average PSNR degradation of the

proposed method is about 0.37dB with the peak value of 0.97dB for the bit rate of 384 kb/s, while it is about 0.30dB with the peak value of 0.9dB for the bit rate of 256 kb/s. The PSNR degradation in the proposed method is mainly caused by the low resolution MC in the down-scale decoder. However, the visual quality of the MPEG4 video obtained by both video transcoders is hardly distinguishable.

5. CONCLUSION AND FUTURE WORK

In this work, we proposed a fast and memory efficient DCT-domain video transcoder to convert high quality MPEG2 video bit streams into low bit rate MPEG4 bit streams with low spatial resolution for wireless video access. Compared to pixel-domain approaches, the proposed video transcoder can save more than 50% of required memory. Furthermore, the computational complexity of the proposed method is less than 30% of that required in the existing method. Experimental results show that the video quality achieved by our transcoder is perceptually identical to that obtained by the pixel-domain method.

In the experiments, the original video sequence is encoded with frame based motion compensation and frame DCT coding mode. However, the MPEG2 standard supports both progressive and interlaced video format. In interlaced video, each frame consists of two fields, the top field and the bottom field. The two fields of a frame may be coded separately (field pictures) or coded together as a frame (frame pictures). Both frame pictures and field pictures may be used in a single video sequence. In a frame picture, both frame DCT coding and field DCT coding can be used on a macro-block basis. Therefore, to support generic video transcoding from MPEG2 to MPEG4, the conversion between different coding modes has to be considered.

6. REFERENCES

[1] H. Sun, W. Kwok, and J. W. Zdepski, "Architectures for MPEG compressed bitstream scaling," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 6, pp. 191–199, Apr. 1996.

[2] P. A. A. Assuncao and M. Ghanbari, "A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bit streams," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 8, pp. 953–967, Dec. 1998.

[3] J. Youn, M.-T. Sun, and C.-W. Lin, "Motion vector refinement for high-performance transcoding," *IEEE Trans. on Multimedia*, vol. 1, pp. 30–40, Mar. 1999.

[4] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats," *IEEE Trans. on Multimedia*, vol. 2, pp. 101–110, June 2000.

[5] B. Shen, I. K. Ishwar, and V. Bhaskaran, "Adaptive motion-vector re-sampling for compressed video down-scaling," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 9, pp. 929–936, Sept. 1999.

[6] R. Dugad and N. Ahuja, "A fast scheme for image size change in compressed domain," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 11, pp. 461–474, Apr. 2001.

[7] S. Liu and A. C. Bovik, "Look-up-table based DCT-domain inverse motion compensation," in *Proc. IEEE Int. Conf. Image Proc.*, (Thessaloniki, Greece), Oct. 2001.

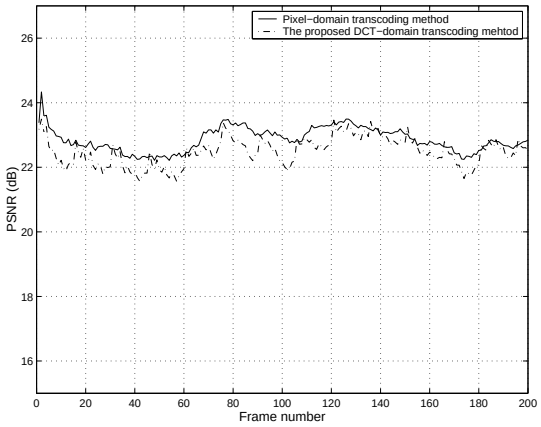

Fig. 5. PSNR values of the MPEG4 video sequence *mobile* encoded at 384 kb/s.

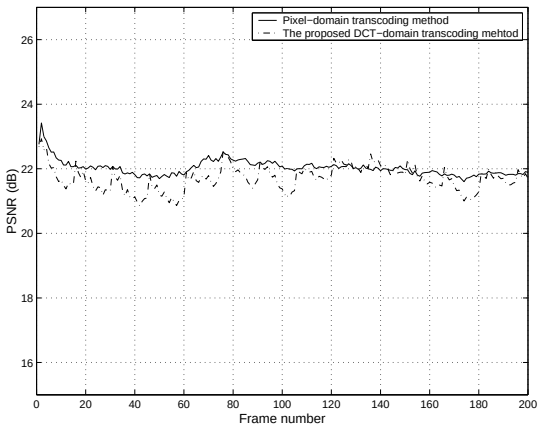

Fig. 6. PSNR values of the MPEG4 video sequence *mobile* encoded at 256 kb/s.