COMPUTATION REDUCTION IN CASCADED DCT-DOMAIN VIDEO DOWNSCALING TRANSCODING

Yuh-Reuy Lee and Chia-Wen Lin
Department of Computer Science & Information Eng.
National Chung Cheng University
Chiayi 621, Taiwan
{lyj,cwlin}@cs.ccu.edu.tw

Yen-Wen Chen
Computer and Communications Research Lab.
Industrial Technology Research Institute
Hisnchu 310, Taiwan
yenwenchen@itri.org.tw

ABSTRACT

In this paper, we propose efficient techniques and architectures for realizing spatial-downscaling transcoders in the DCT domain. We also present methods for re-sampling motion vectors and determining coding modes. We propose a novel drift-free architecture which simplifies the cascaded DCT-domain downscaling transcoder (CDDT) by integrating the downscaling process into the DCT-domain motion compensation (DCT-MC) operation for B frames, thus reducing the computation for DCT-MC and downscaling. We also propose another scheme to further reduce the computation which may introduce drift errors. Experimental results show that the two proposed schemes can achieve significant computation reduction compared with the original CDDT without any degradation or with introducing acceptable quality degradation, respectively.

1. INTRODUCTION

In the recently years, due to the advances of network technologies and wide adoptions of video coding standards, digital video applications become increasingly popular in our daily life. Networked multimedia services, such as video on demand, video streaming, and distance learning, have been emerging in various network environments. These multimedia services usually use pre-encoded videos for transmission. The heterogeneity of present communication networks and user devices poses difficulties in delivering theses bitstreams to the receivers. The sender may need to convert one preencoded bitstream into a lower bit-rate or lower resolution version to fit the available channel bandwidths, the screen display resolutions, or even the processing powers of diverse clients [1]. Many practical applications such as video conversions from DVD to VCD (i.e., MPEG-2 -> MPEG-1) and from MPEG-1/2 to MPEG-4 involve such spatial-resolution, format, and bit-rate conversions. Dynamic bitrate or resolution conversions may be achieved using the scalable coding schemes in current coding standards to support heterogeneous video communications. They, however, usually just provide a very limited support of heterogeneity of bitrates and resolutions (e.g., MPEG-2 and H.263+), or introduce significantly higher complexity at the client decoder (e.g., MPEG-4 FGS).

Video transcoding [1-8] is a process of converting a previously compressed video bit-stream into another bit-stream with a lower bitrate, a different display format (e.g., downscaling), or a different coding method (e.g., the conversion between H.26x and MPEG-x, or adding error resilience), etc. It is considered an efficient means of achieving fine and dynamic adaptation of bitrates, resolutions, and formats. In realizing transcoders, the computational complexity and picture quality are usually the two most important concerns. A straightforward realization of video transcoders is to cascade a decoder followed by an encoder as shown in Fig. 1. This cascaded architecture is flexible and can be used for bitrate adaptation, spatial and temporal resolution-conversion without drift. It is, however, computationally intensive for real-time applications, even though the motion-vectors and coding-modes of the incoming bit-stream can be reused for fast processing.

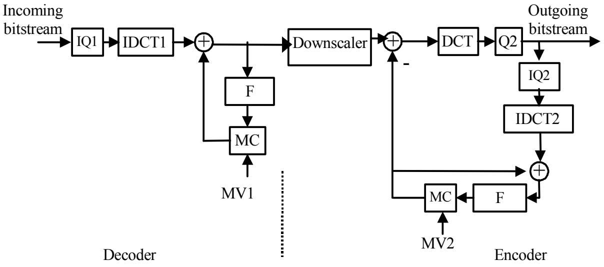


Fig. 1. Cascaded pixel-domain downscaling transcoder (CPDT)

Recently, DCT-domain transcoding schemes [3,4] have become very attractive because they can avoid the DCT and IDCT computations as well as several efficient schemes were developed for implementing the DCT-MC [10-12]. The simplified DCT-domain transcoder proposed in [3], however, cannot be used for spatial/temporal downscaling because it has to use at the encoding stage the same motion vectors decoded from the incoming video. A cascaded DCT-domain downscaling transcoder (CDDT) architecture was first proposed in [4] as depicted in Fig. 2, where a bilinear filtering scheme was used for downscaling the spatial resolution in the DCT domain. A more efficient DCT-domain downscaling scheme, named DCT decimation, was proposed in [5] for image downscaling and later adopted in video transcoding [6]. An architecture similar to the CDDT was proposed in [7], where a reduced-size frame memory is used in the DCT-domain decoder loop for computation and memory reduction which may lead to some drifting errors.

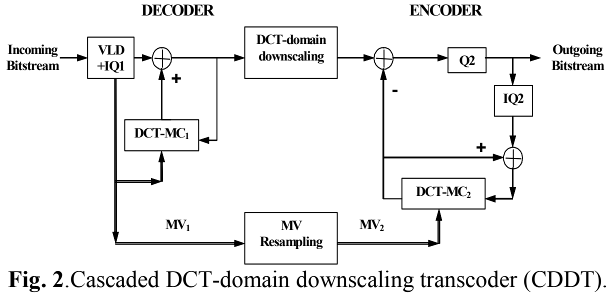


Fig. 2.Cascaded DCT-domain downscaling transcoder (CDDT).

In this paper, we propose efficient architectures for realizing spatial-downscaling transcoders in the DCT domain. We also present methods for re-sampling motion vectors and determining coding modes. We propose a simplified version of CDDT which integrates the downscaling process into the DCT-MC operation for B frames without introducing any degradation. We also propose another scheme to further reduce the computation with acceptable degradation.

2. CASCADED DCT-DOMAIN VIDEO TRANSCODER FOR SPATIAL DOWNSCALING

As mentioned above, the CDDT can avoid the DCT and IDCT computations required in the CPDT as well as preserve the flexibility of changing motion vectors, coding modes as in the CPDT. The major computation required in the CDDT is the DCT-MC operation shown in Fig. 3. It can be interpreted as computing the coefficients of the target DCT block B from the coefficients of its four neighboring DCT blocks, B_i, i = 1 to 4, where B = DCT(**b**) and B_i = DCT($\mathbf{b}_i$) are the 8×8 DCT blocks of the associated pixel blocks **b** and $\mathbf{b}_i$. A close-form solution to computing the DCT coefficients in the DCT-MC operation was firstly proposed in [9] as follows.

$$B = \sum_{i=1}^{4} H_{h_i} B_i H_{w_i} \tag{1}$$

where w_i and $h_i \in \{0,1,\ldots 7\}$. H_{h_i} and H_{w_i} are constant geometric transform matrices defined by the height and width of each sub-block generated by the intersection of $\mathbf{b}_i$ with **b**. Note that, H_{h_i} and H_{w_i} can be pre-computed and then pre-stored in memory. Therefore, no additional DCT computation is required for the computation of Eq. (1).

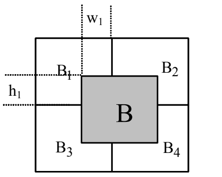


Fig. 3. DCT-domain motion compensation.

A. DCT-Domain Spatial Resolution Downscaling

In [5], an efficient DCT decimation scheme was proposed for spatial downscaling in the DCT domain. This scheme extracts the 4x4 low-frequency DCT coefficients from the four original blocks $\mathbf{b}_1$-$\mathbf{b}_4$, then combines the four 4×4 sub-blocks into an 8×8 block. Let B_1, B_2, B_3, and B_4, represent the four original 8×8 DCT blocks; $\hat{B}_1$, $\hat{B}_2$, $\hat{B}_3$ and $\hat{B}_4$ the four 4×4 low-frequency sub-blocks of B_1, B_2, B_3, and B_4, respectively; $\hat{\mathrm{b}}_i = \mathrm{IDCT}(\hat{B}_i)$, i = 1, …, 4. Then $\hat{\mathbf{b}} = \begin{bmatrix} \hat{\mathbf{b}}_1 & \hat{\mathbf{b}}_2 \\ \hat{\mathbf{b}}_3 & \hat{\mathbf{b}}_4 \end{bmatrix}_{8\times 8}$ is the downscaled version of $\mathbf{b} = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 \\ \mathbf{b}_3 & \mathbf{b}_4 \end{bmatrix}_{16\times 16}$. To compute $\widehat{B} \overset{def}{=} \mathrm{DCT}(\hat{b})$ directly from $\hat{B}_1$, $\hat{B}_2$, $\hat{B}_3$, and $\hat{B}_4$, we can use the following expression:

$$\begin{aligned} \widehat{B} &= T\hat{\mathbf{b}}T^t \\ &= \begin{bmatrix} T_L & T_R \end{bmatrix} \begin{bmatrix} T_4^t \widehat{B}_1 T_4 & T_4^t \widehat{B}_2 T_4 \\ T_4^t \widehat{B}_3 T_4 & T_4^t \widehat{B}_4 T_4 \end{bmatrix} \begin{bmatrix} T_L^t \\ T_R^t \end{bmatrix} \\ &= (T_L T_4^t)\widehat{B}_1 (T_L T_4^t)^t + (T_L T_4^t)\widehat{B}_2 (T_R T_4^t)^t + \\ &\quad (T_R T_4^t)\widehat{B}_3 (T_L T_4^t)^t + (T_R T_4^t)\widehat{B}_4 (T_R T_4^t)^t \end{aligned} \tag{2}$$

In addition to the above formulation, [5] also proposed a decomposition method to convert (2) into a new form so that matrices in the matrix multiplications become more sparse to reduce the computation. This approach was shown to achieve better performance than the filtering schemes.

B. Motion Vector Re-sampling and Mode Decision

After the downscaling, the motion vectors need to be re-sampled to obtain a correct value. Full-range motion re-estimation is computationally too expensive, thus not suited to practical applications. Several methods were proposed for fast re-sampling the motion vectors based on the motion information of the incoming frame [1,6,8,9]. In [1], three motion vector re-sampling methods were compared: median filtering, averaging, and majority voting, where the median filtering scheme was shown to outperform the other two. As a generation of median filtering scheme, we propose to use the activity-weighted median of the four incoming vectors: v_1, v_2, v_3, v_4 proposed in [6,8] as follows:

$$v = \frac{1}{2} \arg \min_{v_i \in \{v_1, v_2, v_3, v_4\}} d_i \tag{3}$$

where the distance measures

$$d_i = \frac{1}{\mathrm{ACT}_i} \sum_{\substack{j=1 \\ j \neq i}}^{4} \left\| v_i - v_j \right\| \tag{4}$$

The macroblock (MB) activity, ACT_i, can be the squared or absolute sum of DCT coefficients, the number of nonzero DCT coefficients, or simply the DC value. In our method, we adopted the squared sum of DCT coefficients of MB as the activity measure.

The MB coding modes also need to be re-determined after the downscaling. In our method, the rules for determining the coding modes are as follows:

(1) If all the four original MBs are intra-coded, then the mode for the downscaled MB is set as intra-coded.

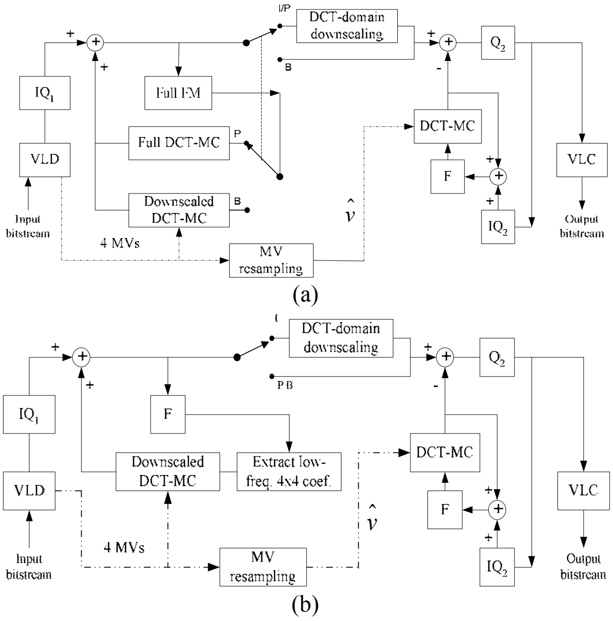
(2) If all the four original MBs are skipped, the resulting downscaled MB will also be skipped.

(3) In all other cases, the mode for the downscaled MB is set as inter-coded.

Note that, the motion vectors of skipped MBs are set to zero.

3. PROPOSED COMPUTATION REDUCTION SCHEMES

We can observe from Fig. 2 that, the decoder-loop of CDDT is operated at the full picture resolution, while the encoding is performed at the quarter resolution. As described in Sec. 2, instead of using the whole DCT coefficients decoded from the decoder loop, the DCT decimation scheme only exploits the 4×4 low-frequency DCT coefficients of each decoded bloc for downscaling. Furthermore, decoding a B-frame with the quarter resolution will not result in any perofrmance degradation in the downscaling transcoder, since B-frames are not used for predicting other frames. A feasible approach for reducing the complexity of CDDT is to perform the full-resolution decoding for I and P frames, and quarter-resolution decoding for B frames by combining the DCT-domain downscaling operation into the DCT-MC of the decoder-loop for B-frames as depicted in Fig. 4(a) (Scheme A). In this way, for B-frames, only the quarter-resolution DCT-MC is required in the decoder-loop, and the DCT-domain downscaling process of B-frames can also be saved. Since B-frames usually occupy a large portion of an I-B-P structured MPEG video, the computation saving can be very significant.



(a)

(b)

Fig. 4. Proposed simplified architectures: (a) Scheme A: only B frames are quarter-resolution decoded; (b) Scheme B: all I, P and B frames are quarter-resolution decoded.

For simplicity, in the following, we show the simplified DCT-MC for decoding B-frames from only one reference frame. It can be easily extended to the case with bidirectional prediction. By incorparating the DCT decimation into the DCT-MC of the decoder-loop for B frames, we obtain

$$\hat{B} = P_4\left(\sum_{i=1}^{4} H_{h_i} B_i H_{w_i}\right) P_4^t \qquad (5)$$

where $P_4 = \begin{bmatrix} I_4 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$, I_4 is an 4×4 identity matrix, and **0** is a 4×4 zero matrix.Then each term in (5) becomes

$$\begin{aligned} P_4 H_{h_i} B_i H_{w_i} P_4^t &= \begin{bmatrix} H_{h_i}^{11} & H_{h_i}^{12} \\ \mathbf{0}_{4\times4} & \mathbf{0}_{4\times4} \end{bmatrix} \begin{bmatrix} B_i^{11} & B_i^{12} \\ B_i^{21} & B_i^{22} \end{bmatrix} \begin{bmatrix} H_{w_i}^{11} & \mathbf{0}_{4\times4} \\ H_{w_i}^{21} & \mathbf{0}_{4\times4} \end{bmatrix} \qquad (6) \\ &= \begin{bmatrix} \left(H_{h_i}^{11} B_i^{11} + H_{h_i}^{12} B_i^{21}\right) H_{w_i}^{11} + & \\ & \mathbf{0}_{4\times4} \\ \left(H_{h_i}^{11} B_i^{12} + H_{h_i}^{12} B_i^{22}\right) H_{w_i}^{21} & \\ \mathbf{0}_{4\times4} & \mathbf{0}_{4\times4} \end{bmatrix} \end{aligned}$$

In (6), the quarter-resolution computation involves 6×4^3 multiplications and 21×4^2 additions, while the corresponding counterpart of (1) needs 2×8^3 multiplications and 14×8^2 additions. Hence the computational complexity of the DCT-MC$_1$ can be reduced significantly. In addition, the computation for DCT-domain downscaling is alos saved. The performance of the simplied archiecture in Fig. 4(a) is exactly the same with the original CDDT.

The computation can be further reduced by applying the quarter-resolution decoding for all P and B-frames (Scheme B in Fig. 4(b)). In this way, each block of the reference P-frame has only 4×4 nonzero low-frequency DCT coefficients (i.e., B^{12}, B^{21}, and B^{22} in (6) are all zero matrices), (5) can thus be reduced as

$$\hat{B} = \sum_{i=1}^{4} \begin{bmatrix} H_{h_i}^{11} B_i^{11} H_{w_i}^{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \qquad (7)$$

However, this simplification will lead to the mismatch between the frame stores of the front-end encoder and the reduced-resolution decoder-loop of the transcoder, thereby resulting in drift errors. The effect of error propagation due to the drift errors will be investigated in the following section.

4. EXPERIMENTAL RESULTS

We compare the performance of the CPDT, the original CDDT, and the proposed schemes A and B. Two test sequences "Football" (with fast motion) and "Flower Garden" (with slow motion) with frame sizes of 720×480 and 704×576, respectively are used for comparison. The two sequences were pre-encoded at 15 Mbps and 30 fps with a GOP structure of (15, 3). The pre-encoded bit-streams are then transcoded into 1.2 Mbps, with downscaled frame-sizes of 352×240 and 352×288, respectively, using the CPDT, CDDT and proposed schemes A and B. The experiments were performed on a Pentium-4 1.8 GHz PC. In order to compare the quality of each scheme, the downscaled videos were up-sampled and interpolated to the original sizes and then compared with the original video. In our experiments, two decimation/interpolation pre-filtering pairs were evaluated for the CPDT: bilinear filtering and a 7-tap filter with the coefficients (-2, 0, 9, 16, 9, 0, -1)/32 suggested in [1], while the CDDT and the proposed methods adopt the DCT-domain decimation and interpolation schemes proposed in [5].

Table 1 compares the average PSNR performance and processing speed of various transcoders. Fig. 5 compares the luminance PSNR values of each frame. The experimental results show that, as compared to the original CDDT, the proposed scheme A can increase the processing speed up to 46% ("Football") and 20% ("Flower-Garden") without any quality degradation for videos with the (15,3) GOP structure. The proposed scheme B can further increase the speed, while introducing about 0.3 dB quality degradation with "Football" and 0.1 dB degradation with "Flower-Garden" in the luminance component. The speed-up gain is dependent on the GOP structure and size used. The larger the number of B-frames in a GOP, the higher the performance gain of proposed scheme A, while the speed-up gain of proposed method B depends on the number of P- and B-frames in a GOP. It is possible to adaptively adopt schemes A and B according to the received block motion information to achieve a better trade-off of speed and video quality.

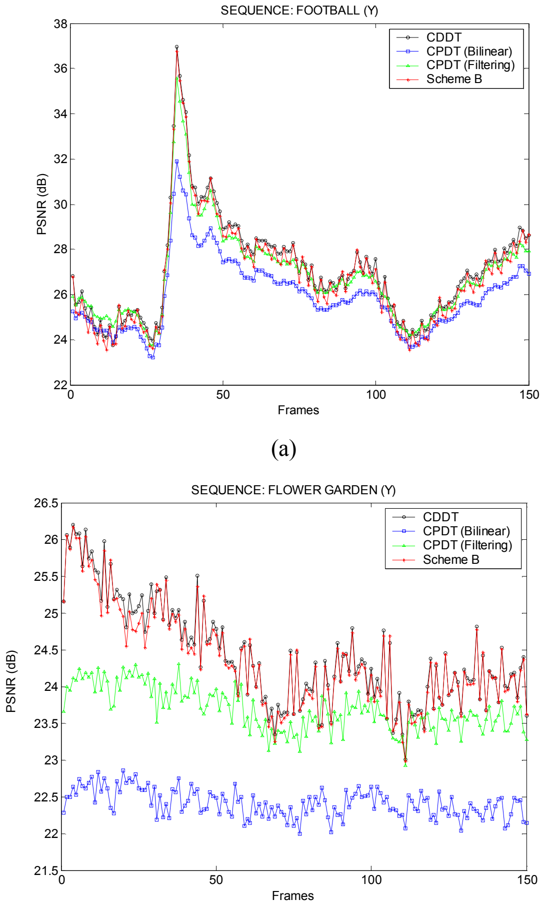
Table 1. Performance comparison of average PSNR and processing speed of CPDT, CDDT and proposed schemes with (a) "Football", and (b) "Flower Garden" sequence.

(a)

Scheme \ Performance		Speed (fps)	Average PSNR (dB)		
			Y	C_r	C_b
CPDT	Bilinear	8.9	25.94	30.05	28.42
CPDT	7-tap Filter	8.3	26.83	35.19	37.47
CDDT		5.9	27.11	35.53	36.72
Proposed	A	8.6	27.11	35.53	36.72
	B	9.0	26.85	34.52	36.31

(b)

Scheme \ Performance		Speed (fps)	Average PSNR (dB)		
			Y	C_r	C_b
CPDT	Bilinear	7.5	22.41	28.17	28.15
CPDT	7-tap Filter	6.9	23.69	30.57	33.55
CDDT		5.9	24.43	30.54	33.37
Proposed	A	7.1	24.43	30.54	33.37
	B	7.4	24.37	30.60	33.42



(a)

(b)

Fig. 5. Luminance PSNR comparison of CDDT and proposed schemes with (a) "Football", and (b) "Flower Garden" sequence.

5. CONCLUSIONS

In this paper, we proposed efficient architectures for DCT-domain spatial-downscaling video transcoders. We have presented an activity-weighted median filtering scheme for re-sampling motion vectors, and a method for determining the coding modes. We have also proposed two novel schemes to integrating the DCT-domain decoding and downscaling operations in the downscaling CDDT into a reduced-resolution DTC-MC so as to achieve significant computation reduction. The proposed scheme A can speed up the decoding and downscaling of B-frames without sacrificing the visual quality, while scheme B can speed up the decoding and downscaling of P- and B-frames with acceptable quality degradation. The proposed schemes can achieve better visual quality while keeping close computational cost as compared to the CPDT. Note, the computation reduction methods for DCT-MC proposed in [1,11,12] can also be used additively to achieve further speed-up.

6. REFERENCES

[1] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats," *IEEE Trans. on Multimedia*, vol. 2, no. 2, pp. 101-110, Jun. 2000.

[2] H. Sun, W. Kwok, and J. W. Zdepski, "Architecture for MPEG compressed bitstream scaling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 2, pp. 191-199, Apr. 1996.

[3] P. A. A. Assuncao and M. Ghanbari, "A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bit streams," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 8, pp. 953-967, Dec. 1998.

[4] W. Zhu, K. Yang, and M. Beacken, "CIF-to-QCIF video bitstream down-conversion in the DCT domain," *Bell Labs technical journal* vol. 3, no. 3, pp. 21-29, Jul.-Sep. 1998.

[5] R. Dugad and N. Ahuja, "A fast scheme for image size change in the compressed domain," *IEEE Trans. Circuit Syst. Video Technol.*, vol. 11, no. 4, pp. 461-474, Apr. 2001.

[6] Y.-R. Lee, C.-W. Lin, and C.-C. Kao, "A DCT-domain video transcoder for spatial resolution downconversion," in *Lecture Notes in Computer Science: Recent Advances in Visual Information Systems*, pp. 207-218, Mar. 2002.

[7] S. Liu and A. C. Bovik, "A fast and memory efficient video transcoder for low bit rate wireless communications," in *Proc. IEEE ICASSP,* pp. 1969-1972, May 2002, Orlando, FL.

[8] J. Xin, M.T. Sun, K. Chun, and B.S. Choi, "Motion re-estimation for HDTV to SDTV transcoding", in *Proc. IEEE ISCAS*, pp. 715-718, May 2002, Arizona.

[9] P. Yin, M. Wu, and B. Liu, "Video transcoding by reducing spatial resolution," in *Proc. IEEE ICIP,* pp. 972-975, Sep. 2000, Vancouver, Canada.

[10] S. F. Chang and D. G. Messerschmitt, "Manipulation and compositing of MC-DCT compressed video," *IEEE J. Select. Areas Commun.*, vol. 13, no. 1, pp. 1-11, Jan. 1995.

[11] C.-W. Lin and Y.-R. Lee, "Fast algorithms for DCT domain video transcoding," in *Proc. ICIP*, pp. 421-424, Oct. 2001, Thessaloniki, Greece.

[12] J. Song and B.-L. Yeo, "A fast algorithm for DCT-domain inverse motion compensation based on shared information in a macroblock," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 5, pp. 767-775, Aug. 2000.