# FAST ALGORITHMS FOR DCT-DOMAIN VIDEO TRANSCODING

Chia-Wen Lin and Yuh-Reuy Lee
Department of Computer Science and Information Eng.
National Chung Cheng University
Chiayi, Taiwan 621, R.O.C.
{cwlin, lyr89}@cs.ccu.edu.tw

## ABSTRACT

Video transcoding is an efficient way for rate adaptation and format conversion in various networked video applications. Many transcoder architectures have been proposed to achieve fast processing. Recently, thanks to its relatively low complexity and acceptable quality, the DCT-domain transcoder (DDT) was proposed to overcome the drift in the open-loop transcoders and the high-complexity in the cascaded transcoders. In this paper, we show that there is still some improvement space for computation reduction in the DDT. We propose a method to fast extract partial low-frequency coefficients in the DCT-domain motion compensation (DCT-MC) operation. We investigate fast algorithms of the DDT based on the proposed fast coefficients extraction scheme. The simulation results show that the proposed methods can achieve significant computation reduction while maintaining close PSNR performance compared to the DDT.

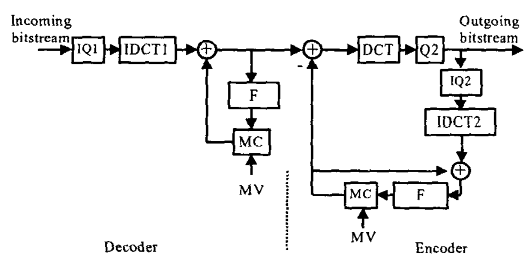
## 1. INTRODUCTION

With the rapid advance of multimedia and networking technologies, multimedia services, such as teleconferencing, video-on-demand, and distance learning have become more and more popular in our daily life. In these applications, it is often needed to adapt the bit-rate of a coded video bit-stream to the available bandwidth over heterogeneous network environments [1]. Dynamic bit-rate conversions can be achieved using the scalable coding schemes provided in current video coding standards [2]. However, it can only provide a limited number of levels of scalability (say, up to three levels in the MPEG standards) of video quality, due to the limit on the number of enhancement layers. In many networked multimedia applications, a much finer scaling capability is desirable. Recently, fine-granular scalable (FGS) coding schemes have been proposed in the MPEG-4 standard to support a fine bit-rate adaptation [3]. However, the video decoder requires additional functionality to decode the enhancement layers in the FGS encoded bit-streams.

Video transcoding is a process of converting a previously compressed video bit-stream into another bit-stream with a lower bit-rate, a different display format (e.g., downscaling), or a different coding method (e.g., the conversion between H.26x and MPEGx, or adding error resilience), etc. In the application of rate adaptation (usually rate reduction), it can provide fine and dynamic adjustments of the bit-rate of the video bit-stream according to the real network conditions without additional functional requirements in the decoder. In realizing transcoders, the computational complexity and picture quality are usually the two most important concerns and need to be traded off to meet various requirements in practical applications. The computational complexity is very critical in real-time applications. For efficient realization of video transcoders, several fast architectures have been proposed in the literature [4-9]. The DCT-domain transcoder (DDT) proposed in [7] is considered competitive due to its relatively low-complexity and low drift when compared to the pixel-domain approaches in [6] and the open-loop methods in [4-5]. The fast cascaded pixel-domain transcoder proposed in [9] has similar computational complexity and higher flexibility with no drift compared to the DDT, while one additional frame memory is required.

In this paper, we propose a fast algorithm to extract partial DCT coefficients in the DCT-MC operation which is the core of the DDT. Based on the proposed fast partial DCT-MC algorithm, we present new methods to speed up the computation of the DDT, while maintaining close picture quality.

The rest of this paper is organized as follows. In section 2, we review existing transcoder architectures, especially the DDT. In section 3, we investigate methods to reduce the computational complexity of the DDT without introducing severe quality degradation. Section 4 compares the performance of the proposed method with the DDT. Finally, a conclusion is provided in section 5.




DCT : Discrete Cosine Transform
IDCT : Inverse Discrete Cosine Transform
Q : Qunatization
MV: Motion Vector
IQ : Inverse Quantization
F : Frame Memory
MC : Motion Compensation

Fig. 1. Cascaded pixel-domain transcoder.

## 2. ARCHITECTURES FOR VIDEO TRANSCODING

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 bit-rate adaptation and spatial and temporal resolution-conversion without drift [4]. It is, however, very 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. As mentioned above, several fast architecture have been proposed for efficient realization of video transcoders [4-9]. In [6], a simplified pixel-domain transcoder (SPDT) was proposed to reduce the computational complexity of the cascade transcoder by reusing motion vectors and merging the decoding and encoding process and eliminating the IDCT and MC (Motion Compensation) operations.

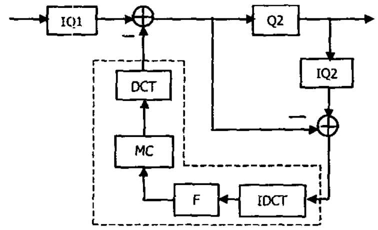



Fig. 2. Simplified pixel-domain transcoder (SPDT).

Further simplifications were proposed by performing the motion-compensation in the DCT-domain [7,10] so that no DCT/IDCT operation is required. In Fig. 2, the functional blocks enclosed by the broken lines can be substituted with a DCT-MC block. The resultant DDT architecture is shown in Fig. 3(a). As shown in Fig. 3(b), the MC-DCT operation 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(**b**$_i$) are the 8×8 blocks of the DCT coefficients of the associated pixel-domain blocks **b** and **b**$_i$ of the image data. It has been shown that the DDT in Fig. 3(a) is mathematically equivalent to the cascaded architecture in Fig. 1 (with motion vector reuse) and the SPDT in Fig.2 [7].

A close-form solution to computing the DCT coefficients in the DCT-MC operation was firstly proposed in [10] as follows.

$$B = \sum_{i=1}^{4} H_{h_i} B_i H_{w_i} \qquad (1)$$

where $w_i$ and $h_i \in \{1,2,\dots 7\}$. $H_{h_i}$ and $H_{w_i}$ are constant geometric transform matrices defined by the height and width of each subblock generated by the intersection of **b**$_i$ with **b**. Direct computation of (1) requires 8 matrix multiplications and 3 matrix additions. Note that, the following relationship holds for the geometric transform matrices: $H_{h_1} = H_{h_2}$, $H_{h_3} = H_{h_4}$, $H_{w_1} = H_{w_3}$, and $H_{w_2} = H_{w_4}$. Using this relationship, the number of operations in (1) can be reduced to 6 matrix multiplications and 3 matrix additions. Moreover, since $H_{h_i}$ and $H_{w_i}$ are deterministic, they can be pre-computed and then pre-stored in memory. Therefore, no additional DCT computation is required for the computation of (1).

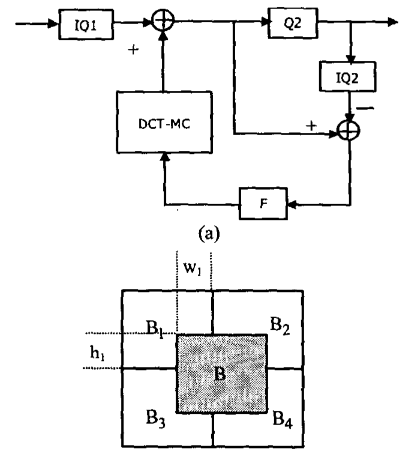



Fig. 3. (a) DCT-domain Transcoder (DDT); (b) motion compensation in the DCT-domain.

## 3. PROPOSED FAST ALGORITHM FOR DCT-DOMAIN TRANSCODING

Since the DCT coefficients stored in the frame memory are the second-stage quantization errors of the sum of the incoming DCT coefficients and the motion-compensated DCT coefficients in the feedback-loop, the energy distributions of the DCT block obtained from the DCT-MC will very likely be mainly concentrated on the low-frequency region. Therefore, for each 8x8 DCT block, usually only few low-frequency coefficients are significant. Thus it would be reasonably accurate to approximate the whole DCT block by calculating only these significant low-frequency coefficients, thereby achieving computation reduction since the number of the significant coefficients is usually small. The number of DCT coefficients need to be calculated to obtain a good approximation of the target block $B$ can be estimated from the energy distributions of the four neighboring anchor blocks $B_1$-$B_4$. In the following, we present a method of fast DCT-MC.

### 3.1. Fast extraction of low-frequency coefficients in DCT-MC

Assume that the number of significant low-frequency DCT coefficients of the $i$th neighboring anchor block $B_i$ is $n_i \times n_i$, to extract $n \times n$ lowest-frequency DCT coefficients of the target block $B$, (1) can be approximated as follows.

$$\hat{B} = T\left(\sum_{i=1}^{4} H_{h_i} T_i B_i T_i H_{w_i}\right) T \qquad (2)$$

where $T = \begin{bmatrix} I_n & 0 \\ 0 & 0 \end{bmatrix}$ and $T_i = \begin{bmatrix} I_{n_i} & 0 \\ 0 & 0 \end{bmatrix}$, $i$=1 to 4

$I_n$ is an $n \times n$ identity matrix and $I_{n_i}$ is an $n_i \times n_i$ identity matrix where $n$ and $n_i$ take values from 0 to 8.

The matrices $B_i$, $H_{h_i}$, and $H_{w_i}$ can be represented as

$$B_i = \begin{bmatrix} B_i^{11} & B_i^{12} \\ B_i^{21} & B_i^{22} \end{bmatrix}, \quad H_{h_i} = \begin{bmatrix} H_{h_i}^{11} & H_{h_i}^{12} \\ H_{h_i}^{21} & H_{h_i}^{22} \end{bmatrix}, \text{ and}$$

$$H_{w_i} = \begin{bmatrix} H_{w_i}^{11} & H_{w_i}^{12} \\ H_{w_i}^{21} & H_{w_i}^{22} \end{bmatrix},$$

where the sub-matrices $H_{h_i}^{11}$, $B_i^{11}$, and $H_{w_i}^{11}$ are of sizes $n \times n_i$, $n_i \times n_i$ and $n_i \times n$ respectively; the sub-matrices $H_{h_i}^{12}$, $B_i^{12}$, and $H_{w_i}^{12}$ are of sizes $n \times (8-n_i)$, $n_i \times (8-n_i)$, and $n_i \times (8-n)$ respectively; the sub-matrices $H_{h_i}^{21}$, $B_i^{21}$, and $H_{w_i}^{21}$ are of sizes $(8-n) \times n_i$, $(8-n_i) \times n_i$ and $(8-n_i) \times n$ respectively; the sub-matrices $H_{h_i}^{22}$, $B_i^{22}$, and $H_{w_i}^{22}$ are of sizes $(8-n) \times (8-n_i)$, $(8-n_i) \times (8-n_i)$ and $(8-n_i) \times (8-n)$ respectively; then each term in (2) becomes

$$TH_{h_i}T_iB_iT_iH_{w_i}T = \begin{bmatrix} I_n & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} H_{h_i}^{11} & H_{h_i}^{12} \\ H_{h_i}^{21} & H_{h_i}^{22} \end{bmatrix} \begin{bmatrix} B_i^{11} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} H_{w_i}^{11} & H_{w_i}^{12} \\ H_{w_i}^{21} & H_{w_i}^{22} \end{bmatrix} \begin{bmatrix} I_n & 0 \\ 0 & 0 \end{bmatrix} \quad (3)$$

$$= \begin{bmatrix} I_n & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} H_{h_i}^{11}B_i^{11}H_{w_i}^{11} & H_{h_i}^{11}B_i^{11}H_{w_i}^{12} \\ H_{h_i}^{21}B_i^{11}H_{w_i}^{11} & H_{h_i}^{21}B_i^{11}H_{w_i}^{12} \end{bmatrix} \begin{bmatrix} I_n & 0 \\ 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} H_{h_i}^{11}B_i^{11}H_{w_i}^{11} & 0 \\ 0 & 0 \end{bmatrix}$$

Substituting (4) into (2), we obtain

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

The number of multiplications and additions required for (3) are $nn_i^2 + n^2n_i$ and $2nn_i^2 + 2n^2n_i - n^2 - nn_i$, respectively. Thus adjusting the numbers $n$, $n_i$ can effectively control the trad-off between the computational complexity and the picture quality, thereby making it computationally scalable. The average number of multiplication and addition operations required for computing (4 for each 8×8 block are thus $2\sum_{i=1}^{4}\sum_{n,n_i} P_nP_{n_i}\left(nn_i^2 + n^2n_i\right)$ and $2\sum_{i=1}^{4}\sum_{n,n_i} P_nP_{n_i}\left(nn_i^2 + n^2n_i - nn_i - n^2\right)$ respectively, where $P_n$ and $P_{n_i}$ represent the probabilities of $n$ and $n_i$ taking values from 0 to 8. On the other hand, the simplified pixel-domain video transcoder depicted in Fig. 2. requires one 8×8 DCT, one 8×8 IDCT, and one block shift operations for each 8×8 block.

### 3.2 Computation reduction in DCT-MC

As mentioned above, the number of sigificant low-frequency DCT coeffcients of the target block $B$ can be estimated from the energy distributions of the four neighboring anchor blocks $B_1$-$B_4$. Define the energy of $N \times N$ coefficients as

$$\text{energy}_i(N) = \sum_{l=0}^{N-1}\sum_{m=0}^{N-1} B_i^2(l,m) \quad (5)$$

where $B_i(l,m)$ is the $l$th row and $j$th column DCT coefficient of $B_i$. $B_i(0,0)$ represents the DC component. For the $i$th neighboring anchor block $B_i$, the associated number of significant coeffcients, $n_i \times n_i$, is determined by calculating the smallest $n_i$ that energy($n_i$)/energy(8) $\geq T$. The threshold value $T$ is empercally set as 0.85, 0.9, and 0.95, respectively in our experiments.

The proposed fast DCT-MC algorithm is as follows:

**Step 1**: For each inter-coded block, estimate $n_i$ for $B_i$, by calculating the smallest $n_i$ that energy($n_i$)/energy(8) $\geq T$, $i$ = 1 to 4. Calculate the overlapping area $k_i$ of the target block $B$ with four neighboring anchor blocks $B_i$, $i$ = 1 to 4.

**Step 2**: ***if*** the largest overlapping area $k_i$ is greater than a predetermined threshold $T$

Chose the neighboring anchor block $B_i$ with the largest overlapping area as the dominant block, and set $n = n_i$

***else***

Estimate $n$ from $n_i$, $i$ = 1 to 4, by using the following bilinear interpolation method:

$$n = \frac{1}{64}\sum_{i=0}^{4} k_i n_i \quad (6)$$

**Step 3**: Compute

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

**Step 4**: Repeat Steps 1-3 until all the inter-coded blocks are processed

The threshold T was set as 0.85×64 (the block area) in our experiments. Note that, the computational complexity of (4) can be further reduced by adopting the method used in [7] in which the elements of the matrices $H_{h_i}$ and $H_{w_i}$ are approximated by using binary numbers with a maximum distortion of 1/32. With this approximation, the multiplications are simplified to basic interger operations, such as *shift-right* and additions (*add*). The shared-information metod proposed in [11] can also be combined with the proposed methods for further speed-up.

## 4. EXPERIMENTAL RESULTS

In our experiments, two QCIF (176x144) image sequences: "Carphone" and "Foreman" with a frame-rate of 10 fps (reduced from a capture frame-rate of 30 fps) were first encoded at 128 Kbps as the test input bit-streams. The test bit-streams are then transcoded into 32 Kbps respectively, using the DDT and FDDT schemes.

Table 1 and Fig. 4 respectively compare the per-frame and average PSNR performance of the proposed FDDT methods with the DDT for threshold $T$ = 0.85, 0.9, and 0.95, respectively. The performance of the proposed FDDT method is very close to the DDT when the threshold is chosen large enough.

Table 1 also compares the measured processing frame-rates of the FDDT and DDT methods. The simulations were performed on an Intel Pentium-III 733 MHz PC. The gain on the overall processing speed using the proposed fast DCT-MC schemes ranges from 12% to 43%. Threshold $T = 0.95$ can achieve good speed-up, while maintaining close quality to the DDT.

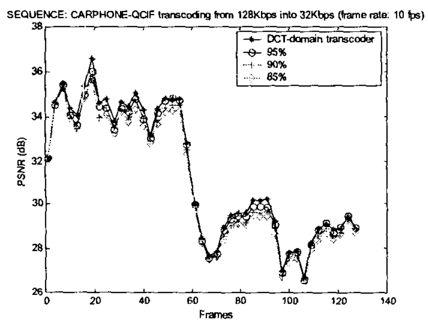
Table 1. Performance comparison of the DDT and two FDDT schemes for three test sequences. The bit-streams are transcoded from 128 Kbps into 32 Kbps. (a) Carphone; (b) Foreman

(a) Carphone

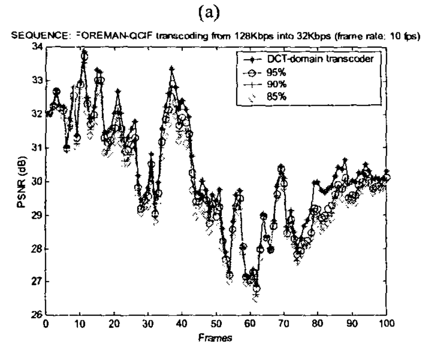
| | DDT | | | FDDT | | | |
|---|---|---|---|---|---|---|---|
| Measured frame rate | 8.98 fps | | | 95% | 10.01 fps | | |
| | | | | 90% | 10.02 fps | | |
| | | | | 85% | 10.06 fps | | |
| | Y | Cr | Cb | | Y | Cr | Cb |
| PSNR (dB) | 31.44 | 37.93 | 39.18 | 95% | 31.28 | 37.06 | 38.64 |
| | | | | 90% | 31.12 | 36.50 | 38.27 |
| | | | | 85% | 30.98 | 36.37 | 38.26 |

(a) Foreman

| | DDT | | | FDDT | | | |
|---|---|---|---|---|---|---|---|
| Measured frame rate | 6.98 fps | | | 95% | 9.95 fps | | |
| | | | | 90% | 9.97 fps | | |
| | | | | 85% | 9.98 fps | | |
| | Y | Cr | Cb | | Y | Cr | Cb |
| PSNR (dB) | 30.36 | 38.78 | 38.10 | 95% | 30.05 | 37.57 | 37.67 |
| | | | | 90% | 29.95 | 37.57 | 37.31 |
| | | | | 85% | 29.79 | 37.44 | 37.36 |




(a)




(b)

Fig. 4. PSNR comparison of three DCT-domain transcoding schemes for (a) "Akiyo" sequence; (b) "News" sequence. The bit-streams are transcoded from 128 Kbps into 32 Kbps.

## 5. CONCLUSION

In this paper, we proposed a fast algorithm for extracting partial low-frequency coefficients of the target DCT blocks in the DCT-MC operation. We also proposed efficient schemes to estimate the number of significant coefficients containing most of the energy of the target DCT block in the DCT-MC operation. We showed that, by combining with the proposed schemes, the computational complexity of the DCT-MC can be significantly reduced without introducing severe quality degradation. The other advantage of the proposed method is its computation scalability.

## 6 REFERENCES


[1] J. Moura, R. Jasinschi, H. Shiojiri-H, and C. Lin, "Scalable video coding over heterogeneous networks," in *Proc. SPIE*. vol. 2602, pp. 294-306. 1996.

[2] M. Ghanbari, "Two-layer coding of video signals for VBR networks," *IEEE J. Select. Areas Commun.*, vol.7, pp.771-781, June 1989.

[3] W. Li, "Fine-Granular scalability using bit-plane coding of DCT coefficients", ISO/IEC MPEG Document M4204, Roma, IT, DEC., 1998..

[4] H. Sun, W. Kwok and J.W. Zdepski, "Architecture for MPEG compressed bitstream scaling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 2, pp191-199, Apr. 1996.

[5] A. Eleftheriadis and D. Anastassiou, "Constrained and general dynamic rate shaping of compressed digital video," in *Proc. IEEE Int. Conf. Image Processing*, Washington DC, Oct. 1995.

[6] G. Keesman, *et al.*, "Transcoding of MPEG bitstreams," *Signal Processing: Image Commun.*, vol. 8, pp. 481-500, 1996.

[7] 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, Dec. 1998.

[8] J. Youn, M.T. Sun, and J. Xin, "Video transcoding architectures for bit rate scaling of H.263 bit streams," in *Proc. ACM Multimedia Conf.* 1999, Orlando, Nov. 1999.

[9] J. Youn, J. Xin, and M.T. Sun, "Fast video transcoder architecture for networked multimedia applications," in *Proc. IEEE Int. Symp. Circuits and Systems*, Geneva, Switzerland, May 2000.

[10] S. F. Chang and D. G. Messerschmitt, "Manipulation and compositing of MC-DCT compressed video," *IEEE J. Select. Areas Commun.*, pp. 1-11, Jan. 1995.

[11] 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.