DCT-DOMAIN SPATIAL TRANSCODING USING GENERALIZED DCT DECIMATION

Yuh-Ruey Lee and Chia-Wen Lin

Department of Computer Science and Information Engineering
National Chung Cheng University, ROC
Chiayi, Taiwan 621, ROC
{lyj, cwlin}@cs.ccu.edu.tw

ABSTRACT

In this paper, we propose a generalized DCT-domain spatial downscaling scheme to improve the visual quality. We analyze the filtering performances and computational complexities of the proposed scheme and the pixel-domain downscaling schemes. The analyses show that the proposed scheme can reduce the aliasing artifact compared to the existing schemes, while the computational complexity may be increased. We also integrate the proposed decimation scheme into the cascaded DCT-domain transcoder for spatial downscaling of a pre-encoded video into its quarter size. Experiments show the proposed approach can achieve better visual quality than the existing schemes.

1. INTRODUCTION

In recently years, due to the advances of network technologies and wide adoptions of video coding standards, digital video applications have 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 these bitstreams to the receivers. The sender may need to convert one pre-encoded 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].

Video transcoding [1]-[3] is an operation of converting a video bit-stream into from one format into another format (e.g., bit-rate, frame-rate, spatial resolution, and coding syntax). It is an efficient means of achieving fine and dynamic video adaptation. 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. 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.

Recently, DCT-domain transcoding schemes [3] have become very attractive because they can avoid the DCT and IDCT computations as well as several efficient schemes have been developed for implementing the DCT-domain motion compensation (DCT-MC) [4]. A cascaded DCT-domain downscaling transcoder (CDDT) architecture was first proposed in [5] as depicted in Fig. 1, where a bilinear filtering scheme was used for the spatial resolution downscaling in the DCT domain.

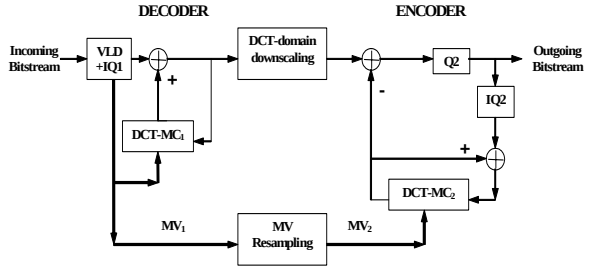


Fig. 1. Cascaded DCT-domain downscaling transcoder (CDDT).

An efficient DCT-domain downscaling scheme, namely, DCT decimation, was proposed in [6] for image downscaling and later adopted in [7] for video transcoding. The DCT decimation scheme first extracts the 4×4 low-frequency DCT coefficients of the four original blocks $\mathbf{b}_1$~$\mathbf{b}_4$, and then combines the four 4×4 sub-blocks into an 8×8 block. Let B_1~B_4, represent the four original 8×8 DCT blocks; $\hat{B}_1 \sim \hat{B}_4$ the four 4×4 low-frequency sub-blocks of B_1~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 a downscaled version of $\mathbf{b} = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 \\ \mathbf{b}_3 & \mathbf{b}_4 \end{bmatrix}_{16\times 16}$.

The following formula can be used to compute $\hat{B} = \mathrm{DCT}(\hat{\mathbf{b}})$ [6].

$$\hat{B} = T\hat{\mathbf{b}}T^t = \begin{bmatrix} T_L & T_R \end{bmatrix} \begin{bmatrix} T_4^t \hat{B}_1 T_4 & T_4^t \hat{B}_2 T_4 \\ T_4^t \hat{B}_3 T_4 & T_4^t \hat{B}_4 T_4 \end{bmatrix} \begin{bmatrix} T_L^t \\ T_R^t \end{bmatrix} \quad (1)$$

The DCT decimation scheme with the 8×8 block size has proven to achieve better visual quality than schemes using pixel-domain filtering followed by subsampling [6][7]. However, it may still lead to blocky artifacts due to that only the low-frequency 4×4 DCT coefficients of a block are kept whereas the others are discarded. The blocky artifacts often arise in complex texture regions in which the high-frequency DCT coefficients are much more significant than those in smooth regions. The more the high-frequency coefficients of a block are discarded, the more visible the blocky artifacts in the downscaled image. To reduce the blocky artifacts at block boundaries, we can use larger than 8×8 DCT block sizes for spatial downscaling though the computational complexity will increase. In this paper, we propose a generalized DCT-domain decimation scheme which performs decimation by two on sub-frames of a flexible size

other than the traditional 8×8 block size used in [6]. We shall analyze the performance and complexity of the proposed generalized DCT decimation scheme. In addition, we also integrate the proposed method into the CDDT architecture shown in Fig. 1 and compare the performance.

2. GENERALIZED DCT-DOMAIN SPATIAL RESOLUTION DOWNSCALING

In this section, we present a generalized DCT-domain decimation-by-two scheme. In our scheme, a coded video frame is first decoded into a pixel image by 8×8 IDCT and then divided into sub-frames of the size of $N\times M$ points (N denotes the vertical size and M denotes the horizontal size, which are multiples of eight) rather than 8×8 blocks used in [6]. Each $N\times M$ sub-frame is transformed into its corresponding DCT sub-frame by $N\times M$-point DCT. The DCT decimation is then performed on each $N\times M$ DCT sub-frame by extracting only the $(N/2)\times(M/2)$ low-frequency DCT coefficients, and then transforming these retained coefficients back to a downscaled version by $(N/2)\times(M/2)$-point IDCT. Finally each downscaled sub-frame is encoded into 8×8 DCT blocks to form the output video. The above procedures can be combined together and performed fully in the DCT domain as described in the following three steps:

Step 1. Grouping a set of 8×8 blocks to an N×M DCT sub-frame

The IDCT transformation of each 8×8 DCT block $B_{i,j}$ in the original-size frame is expressed as

$$\mathbf{b}_{i,j} = T_8^t B_{i,j} T_8, \quad i = 1,..., N/8 \text{ and } j = 1,\ldots, M/8 \tag{2}$$

where $\mathbf{b}_{i,j}$ and $B_{i,j}$ are the (i,j)th 8×8 pixel block and the corresponding 8×8 DCT block, respectively. T_8 is the 8-point 1-D DCT transform matrix.

Suppose the frame is divided into sub-frames of $N\times M$ pixels. Let T_N and T_M denote the N-point and M-point DCT transform matrices, respectively. The $N\times M$-sized DCT sub-frame $F_{N\times M}$ can be computed from the corresponding 8×8 DCT blocks by

$$\begin{aligned} F_{N\times M} &= T_N \begin{bmatrix} \mathbf{b}_{1,1} & \cdots & \mathbf{b}_{1,M/8} \\ \vdots & \ddots & \vdots \\ \mathbf{b}_{N/8,1} & \cdots & \mathbf{b}_{N/8,M/8} \end{bmatrix} T_M^t \\ &= \begin{bmatrix} T_{N,1} & \cdots & T_{N,N/8} \end{bmatrix} \begin{bmatrix} T_8^t B_{1,1} T_8 & \cdots & T_8^t B_{1,M/8} T_8 \\ \vdots & \ddots & \vdots \\ T_8^t B_{N/8,1} T_8 & \cdots & T_8^t B_{N/8,M/8} T_8 \end{bmatrix} \begin{bmatrix} T_{M,1}^t \\ \vdots \\ T_{M,M/8}^t \end{bmatrix} \end{aligned} \tag{3}$$

In (3), T_N is divided into $N/8$ columns of sub-matrices $T_{N,i}$ of size $N\times 8$, while T_M is divided into $M/8$ rows of sub-matrices $T_{M,j}^t$ of size $8\times M$. Equation (3) can be rewritten as follows.

$$\begin{aligned} F_{N\times M} &= \left(T_{N,1}T_8^t\right)B_{1,1} + \cdots + \left(T_{N,N/8}T_8^t\right)B_{N/8,1}\left(T_8 T_{M,1}^t\right) \\ &+ \left(\left(T_{N,1}T_8^t\right)B_{1,2} + \cdots + \left(T_{N,N/8}T_8^t\right)B_{N/8,2}\right)\left(T_8 T_{M,2}^t\right) \\ &+ \ldots \\ &+ \left(\left(T_{N,1}T_8^t\right)B_{1,M/8} + \cdots + \left(T_{N,N/8}T_8^t\right)B_{N/8,M/8}\right)\left(T_8 T_{M,M/8}^t\right) \end{aligned} \tag{4}$$

where $T_{N,i}T_8^t$ and $T_8 T_{M,j}^t$ can be calculated and stored in tables off-line for computing $F_{N\times M}$.

Step 2. Extracting (N/2)×(M/2) low-frequency coefficients

In order to obtain the downscaled version of the full resolution frame, the $(N/2)\times(M/2)$ low-frequency coefficients of (4) are subsequently extracted as expressed in (5).

$$\hat{F}_{(N/2)\times(M/2)} = P_{(N/2)\times N} F_{N\times M} Q_{M\times(M/2)} \tag{5}$$

where $P_{N/(2\times N)} = \begin{bmatrix} I_{(N/2)\times(N/2)} & \mathbf{0}_{(N/2)\times(N/2)} \end{bmatrix}$ and $Q_{M\times M/2} = \begin{bmatrix} I_{(M/2)\times(M/2)} \\ \mathbf{0}_{(M/2)\times(M/2)} \end{bmatrix}$ are matrices for extracting low-frequency coefficients.

Since only the low-frequency coefficients are retained, the remaining high-frequency coefficients are all discarded. Therefore, in (4), only the computations for retained low-frequency coefficients in (5) are required, whereas the others can be saved. The computation can thus be reduced to

$$\begin{aligned} \hat{F}_{N/2\times M/2} &= \left(L_{N,1}B_{1,1} + \cdots + L_{N,N/8}B_{N/8,1}\right)R_{M,1}^t \\ &+ \left(L_{N,1}B_{1,2} + \cdots + L_{N,N/8}B_{N/8,2}\right)R_{M,2}^t \\ &+ \ldots \\ &+ \left(L_{N,1}B_{1,M/8} + \cdots + L_{N,N/8}B_{N/8,M/8}\right)R_{M,M/8}^t \end{aligned} \tag{6}$$

where $L_{N,i} = P_{N/(2\times N)} T_{N,i} T_8^t$ and $R_{M,j}^t = T_8 T_{M,j}^t Q_{M\times(M/2)}$ for $i \in \{1, ..., N/8\}$ and $j \in \{1, ..., M/8\}$. Note that $L_{N,i}$ and $R_{M,j}^t$ can be calculated and stored offline, thus will not consume extra computation while performing transcoding.

Step 3. Converting $\hat{F}_{(N/2)\times(M/2)}$ to 8×8 DCT blocks of the downscaled frame

By performing $(N/2)\times(M/2)$-point IDCT on the downscaled DCT sub-frame, as shown in (6), we can obtain the downscaled pixel-domain sub-frame $\hat{\mathbf{f}}_{(N/2)\times(M/2)}$.

$$\hat{\mathbf{f}}_{\frac{N}{2}\times\frac{M}{2}} = T_{\frac{N}{2}}^t \hat{F}_{(N/2)\times(M/2)} T_{\frac{M}{2}} = \begin{bmatrix} T_{N/2,1}^t \\ \vdots \\ T_{N/2,N/16}^t \end{bmatrix} \hat{F}_{(N/2)\times(M/2)} \begin{bmatrix} T_{\frac{M}{2},1} & \cdots & T_{\frac{M}{2},\frac{M}{16}} \end{bmatrix} \tag{7}$$

According to (7), each 8×8 pixel block in the downscaled sub-frame is computed by $\hat{\mathbf{b}}_{k,l} = T_{N/2,k}^t \hat{F}_{(N/2)\times(M/2)} T_{M/2,l}$, for k = 1, ..., $(N/2)/8$ and l = 1, …, $(M/2)/8$. The corresponding 8×8 DCT block is $\hat{B}_{k,l} = T_8 \hat{\mathbf{b}}_{k,l} T_8^t$.

Therefore, $\hat{B}_{k,l}$ can be computed directly from the downscaled DCT sub-frame $\hat{F}_{(N/2)\times(M/2)}$ by

$$\hat{B}_{k,l} = \left(T_8\, T_{N/2,k}^t\right)\hat{F}_{(N/2)\times(M/2)}\left(T_{M/2,l} T_8^t\right) = S_{N/2,k}^t \hat{F}_{(N/2)\times(M/2)} S_{M/2,l} \tag{8}$$

where $S_{N/2,k}^t = T_8 T_{N/2,k}^t$ and $S_{M/2,k} = T_{N/2,k} T_8^t$ are matrices with sizes of $8\times(N/2)$ and $(M/2)\times 8$, respectively.

The above procedures can be combined together to be performed in the DCT domain as summarized below:

1. *Divide an input coded frame F into sub-frames with size N×M, each consisting of 8×8 DCT blocks $B_{i,j}$ for i = 1, ..., N/8 and j = 1, ..., M/8.*
2. *Use (6) to extract each DCT sub-frame $\hat{F}_{(N/2)\times(M/2)}$.*
3. *Use (8) to calculate each outgoing DCT block, $\hat{B}_{k,l}$, of the downscaled frame, for k = 1, ..., (N/2)/8 and l = 1, …, (M/2)/8 .*

3. ANALYSES OF DCT-DECIMATION DOWNSCALING FILTERS

The operation of retaining the low-frequency coefficients of a DCT sub-frame and taking the half-size IDCT is, in effect, to perform anti-aliasing filtering and then followed by down-sampling on the sub-frame in the pixel domain. This results in a downscaled version of the sub-frame. In the following, we shall analyze the performances and complexities of various downscaling filters for the 1-D case. The same analyses apply to the 2-D case since separable 2-D DCT is used in the decimation. For N samples of 1-D signal $\mathbf{x}$, when downscaled by a factor of two, the downscaled $N/2$-sample signal $\mathbf{y}$ is obtained as follows.

$$\mathbf{y} = T_{N/2}^{t} P_{(N/2)\times N} T_N \mathbf{x} \tag{9}$$

The downscaling filter is defined as

$$M_{N/2\times N} = T_{N/2}^{t} P_{N/2\times N} T_N \tag{10}$$

Using the downscaling filter, the input signal is linearly transformed with matrix $M_{(N/2)\times N}$ to obtain a corresponding down-sampled output signal. The linear transform can be represented as an N-band filter bank structure shown in Fig. 2, where each filter h_i is the reverse order of the ith row of the matrix $M_{(N/2)\times N}$. Hence, the z-transform of the output $\mathbf{y}$ can be obtained by

$$Y(z) = \frac{1}{N}\sum_{k=0}^{N-1} X(W^{-k}\sqrt{z}) F_k(W^{-k}\sqrt{z}) \tag{11}$$

where $W=\exp(j2\pi/N)$, $j=\sqrt{-1}$ and

$$F_k(z) = \sum_{i=0}^{N/2-1} z^{-2i} H_i(z) W^{-2ki} \,. \tag{12}$$

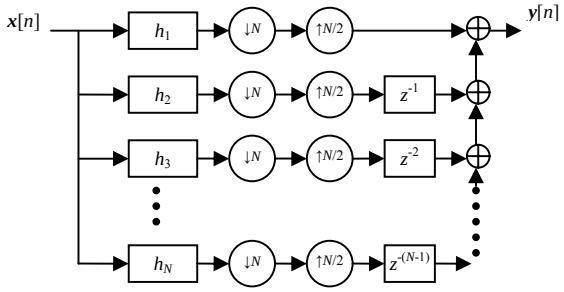


Fig. 2. The filter bank structure represents the downscaling operation.

Fig. 3 shows the magnitude responses of the two pixel-domain filters: the bilinear filter and the 7-tap filter (as suggested in [2]), and the generalized DCT decimation filters with N = 8, 16, 72, and 288, respectively. As N increases, the gain of DCT decimation filters, $|F_0(z)|$, becomes much flatter in the low-frequency part ($0\sim\pi/2$), whereas the gain decreases rapidly in the high-frequency part ($\pi/2\sim\pi$). For the bilinear filter, the gain in the high-frequency part is always larger than its counterparts of DCT decimation filter and 7-tap filter. The smaller gain in the high-frequency part implies less visible aliasing artifacts in the downscaled image. Although increasing the sub-frame size for the DCT decimation filters will lead to better quality of downscaled image, it will also increase the computational complexity significantly. The computational complexities of the proposed DCT-decimation scheme with various sub-frame sizes are listed in Table I. From the table, the 8×8 sub-frame size used in [6] achieves the lowest complexity whereas the other larger sub-frame sizes lead to higher complexities. However, the visual quality of the scheme with a larger sub-frame size is better than that with a smaller sub-frame size as will be compared later.

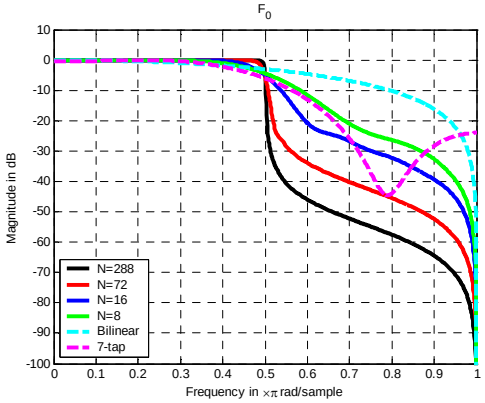


Fig. 3. Comparison of magnitude responses of DCT-domain and pixel-domain decimation filters.

Table I
Average computational complexity per block for each downscaling scheme

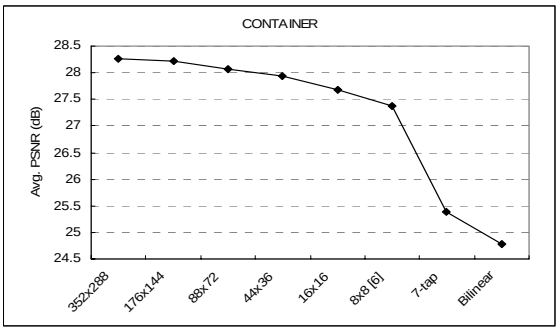
Sub-frame size, $N\times M$	Avg. multiplications per block $\hat{B}_{k,l}$	Avg. additions per block $\hat{B}_{k,l}$
352×288	227,840	224,640
176×144	63,232	61,600
88×72	18,944	18,096
44×36	6,304	5,848
16×16	1,792	1,568
8×8 [6]	1,024	768

4. EXPERIMENTAL RESULTS

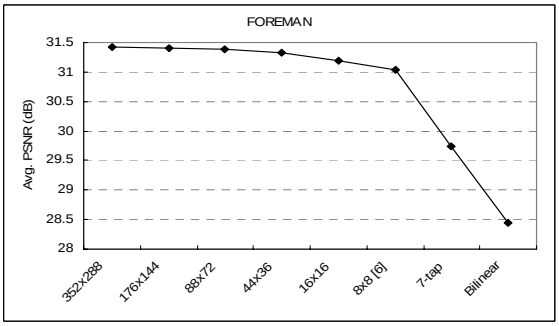
We evaluate the performances of DCT decimation with six different sub-frame sizes. Two 150-frame CIF (352×288) test sequences are encoded by an MPEG-2 encoder at 4 Mbps and 30 fps with the (15,3) GOP structure. The DCT-domain spatial downscaling transcoder performs the proposed downscaling scheme on an incoming coded video, and produces a spatially downscaled video of the QCIF size (176×144) coded at 1 Mbps. For comparing the performances of these schemes, the downscaled bitstreams are decoded and up-scaled to its original size, and then compute the average PSNR with its pre-encoded video. The up-scaling method is, similar to that proposed in [6], the reverse procedure of the DCT decimation schemes as summarized below:

1) Divide each downscaled video frame into sub-frames of $(N/2)\times(M/2)$ pixels. Then transform each pixel-domain sub-frame to a DCT sub-frame by $(N/2)\times(M/2)$ 2-D DCT.
2) Expand the size of each DCT sub-frame by a factor of two in height and width (i.e., $N\times M$) with stuffing of zero coefficients in the high-frequency bands expended.
3) Use $N\times M$ IDCT to transform each expanded DCT sub-frame into pixel-domain sub-frame of $N\times M$ pixels.

Fig. 4 shows the average PSNR comparison of the DCT decimation scheme with six different sub-frame sizes and the pixel-domain downscaling schemes. Fig. 5 shows the frame-by-frame PSNR performance comparison. The larger the sub-frame size, the better the visual quality. The average PSNR performance improvement can be up to 0.9 dB for *Container* and 0.4 dB for *Foreman*. However, the performance improvement comes with an increased computational complexity. The DCT decimation schemes significantly outperform the pixel-domain downscaling filtering (the bilinear filter and the 7-tap filter) schemes in terms of average PSNR by up to 1.7~2.8 dB and 3~3.5 dB, respectively, for the two test sequences.

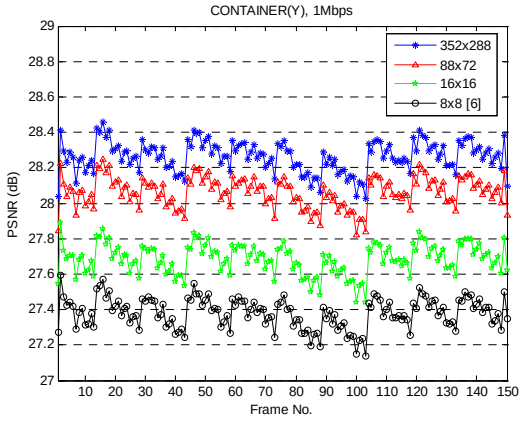


(a)

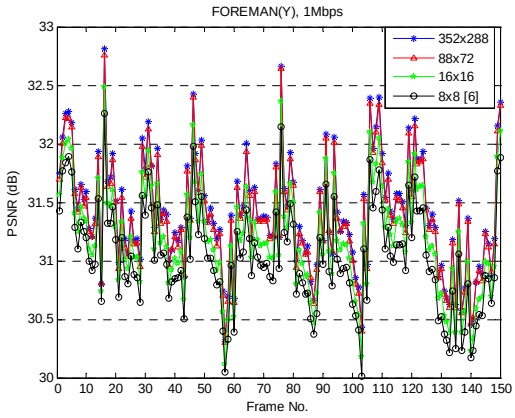


(b)

Fig. 4. Average PSNR performance comparison of the proposed approach with six sub-frame sizes for (a) *Container* and (b) *Foreman*.



(a)



(b)

Fig. 5. Frame-by-frame PSNR performance comparison of the proposed approach with six sub-frame sizes for (a) *Container* and (b) *Foreman*.

5. CONCLUSION

We proposed a generalized DCT decimation scheme which can adopt sub-frame sizes large than the traditional 8×8 block size. We analyzed the anti-aliasing filtering performance of the proposed scheme and the bilinear and 7-tap filtering schemes. We have also implemented a DCT-domain spatial downscaling transcoder based on the proposed scheme. Experiments show that the proposed scheme with a sub-frame size larger than 8×8 can achieve better visual quality, while leading to an increased computational cost. The PSNR performance improvement of using a large sub-frame size can be up to 0.9 dB on average.

REFERENCES

[1] J. Xin, C.-W. Lin, and M.-T. Sun, "Digital video transcoding," *Proc. IEEE*, vol. 93, no. 1, pp. 84-97, Jan. 2005.

[2] 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.

[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] 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.

[5] 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.

[6] 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.

[7] Y.-R. Lee, C.-W. Lin, and Y.-W. Chen, "Computation reduction in cascaded DCT-domain video downscaling transcoder," in *Proc. IEEE ISCAS*, vol. 2, pp. 860-863, May 2003.