Optimal Down-Conversion in Compressed DCT Domain with Minimal Operations

Myoung-Cheol Shin and In-Cheol Park

Dept. of EECS, Korea Advanced Institute of Science and Technology
373-1, Kusong-dong, Yusong-gu, Taejon 305-701, KOREA

ABSTRACT

A new down-conversion scheme in the DCT domain is presented, which can be used in decoders of DCT-compressed images and videos. The down-conversion in the transform domain generally requires lower computational complexity than the spatial domain down-conversion. The proposed method also requires computational complexity comparable to other DCT domain techniques, and it is optimal in MSE sense, whereas the others are not. We minimized the number of arithmetic operations without discarding any data in DCT coefficients. We first combined the DQ, IDCT, and spatial domain averaging together. To minimize the number of operations used, we then concentrated most of the multiplications to the first stage and performed them at once. As a result, the proposed scheme shows better PSNR characteristic than other DCT domain methods while it requires almost the same number of operations and memory.

Keywords: down-conversion, compressed domain processing, DCT, image processing, JPEG, MPEG

1. INTRODUCTION

Image down-conversion, as well as up-conversion and translation, is one of fundamental and indispensable image manipulations. For example, in the very near future, the transmission of high-definition television (HDTV) employing MPEG-2 will spur a transition period in which both HDTV and standard-definition television (SDTV) will be received. To successfully implement such a system, down conversion is necessary to display the HDTV signal on a low-resolution monitor.

Many image and video compression standards such as JPEG, MPEG, and H.261 compress the data using a sequence of 8×8 discrete cosine transform (DCT) and scalar quantization coding techniques. If we manipulate image and video in the compressed DCT domain, we can reduce computational complexity due to much lower data rate in the compressed domain. There have been considerable researches for processing images and videos directly in the DCT domain.[1,2]

Many DCT domain down-conversion schemes have also been proposed so that the decoder system can reconstruct low-resolution images from high-resolution bit streams, while keeping low computational complexity.[3-5] Most of these techniques attempt to extract certain coefficients from an 8×8 DCT block. In the simplest case, the top-left 4×4 low-frequency terms are used to reconstruct a 4×4 image block.[3] We will refer to this scheme as *4×4 Cut*. However, because these techniques are based on zonal low-pass filtering, they generate ringing artifacts around high frequency areas, which degrade the subjective quality of the down-converted image. As a matter of fact, they are not optimal in a sense of mean square error (MSE), i.e. they don't minimize the MSE between original and down-converted images. That is because the energy in discarded high-frequency data results in noise in the spatial domain. Although sophisticated techniques have been developed to preserve high-frequency content,[6,7] they are still suboptimal in MSE sense and require much computational complexity.

In MSE sense, averaging pixel data in spatial domain is the optimal solution. It also shows good subjective quality and produces no artifacts such as ringing. It is commonly used and can be implemented with simple operations, but the sequence of the full 8×8 inverse DCT (IDCT) and the spatial averaging needs a large number of operations.

Correspondence: Email: mcshin@vslab.kaist.ac.kr, icpark@ee.kaist.ac.kr

We propose a down-conversion decoding scheme with the same *minimum MSE* as spatial averaging and a *minimal number of arithmetic operations*. The minimal number of operations is achieved by first combining the IDCT stage and the spatial averaging stage in the context of compressed domain processing, and then by transferring multiplications in the combined transform stages to the scalar dequantization (DQ) stage. Since the DQ stage originally performs multiplication to each DCT coefficient, the transferred multiplications do not increase the number of operations, but only change the values that are multiplied.

This paper is organized as follows: In the following Section 2, we describe our optimal down-conversion scheme. Section 3 presents experimental results and compares our methods with others. Finally, conclusions of this work are given in Section 4.

2. OPTIMAL DOWN CONVERSION

In most compression standards, image and video data are compressed using the sequence of 8×8 DCT and scalar quantization techniques. Scalar dequantization therefore precedes 8×8 IDCT in a decoder system. To reconstruct a down-converted image or video with the minimum MSE, three operations, including scalar dequantization, IDCT, and spatial-domain down conversion, are needed. In a conventional down-conversion decoder, the operations may be implemented in three separate stages as depicted in Figure 1(a). The proposed down-conversion decoder performs the operations in two stages, *scaling* and *transform* as shown in Figure 1(b). Neither of the stages is the same as one in Figure 1(a). We once combined the original three operations together. Then we formed the first scaling stage with all the multiplications that are to be applied to each DCT term, and the transform stage with other inter-term operations.

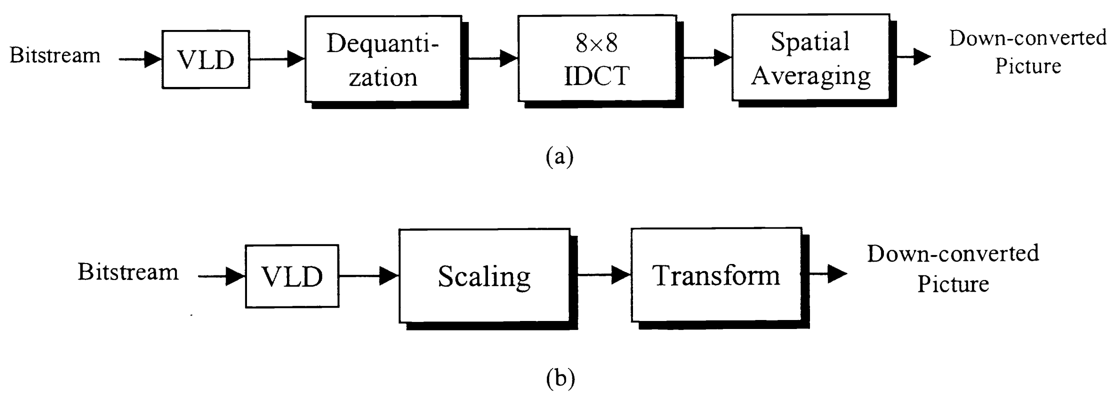

Figure 1: Decoder block diagram. using (a) spatial domain down conversion and (b) proposed scheme.

2. 1 Combining DQ, IDCT, and down-conversion

The DQ stage in Figure 1(a) receives a quantized level and a position of a DCT coefficient from the variable length decoder (VLD). It multiplies the given level by a quantizing step and saves it in the given position in the DCT coefficient matrix. Because the quantizing steps vary according to the position, they are maintained as a matrix form, which is called the quantization matrix. When the DCT coefficient matrix is filled with proper values, it is sent to the IDCT stage.

The IDCT stage immediately follows the DQ. It performs the 8×8 two-dimensional IDCT on the DCT coefficient matrix from DQ to reconstruct an image block. The IDCT stage transforms a matrix of DCT coefficients $\mathbf{X} = \{X(k,l)\}_{k,l=0}^{7}$ into a block in the spatial domain $\mathbf{x} = \{x(n,m)\}_{n,m=0}^{7}$ by using the following equation:

$$x(n,m)=\sum_{k=0}^{7}\sum_{l=0}^{7}\alpha(k)\alpha(l)X(k,l)\cos\left(\frac{2n+1}{16}\cdot k\pi\right)\cos\left(\frac{2m+1}{16}\cdot l\pi\right), \tag{1}$$

where $\alpha(0)=1/2\sqrt{2}$ and $\alpha(k)=1/2$ for $k>0$. Given the one-dimensional eight-point IDCT matrix $\mathbf{C}=\{c(n,k)\}_{n,k=0}^{7}$ where

$$c(n,k)=\alpha(k)\cos\left(\frac{2n+1}{16}\cdot k\pi\right), \tag{2}$$

the relation between $\mathbf{x}$ and $\mathbf{X}$ is represented in matrix equation as

$$\mathbf{x}=\mathbf{CXC}^{t} \tag{3}$$

where the superscript t denotes matrix transposition.

Finally the down-conversion stage performs the 1:2 down-conversion in each dimension using a spatial averaging method. The operation is represented in a matrix form as

$$\mathbf{x}'=\mathbf{AxA}^{t} \tag{4}$$

where $\mathbf{x}'$ is the down-converted 4×4 image matrix, $\mathbf{A}$ is the 4×8 vertical averaging matrix defined as

$$\mathbf{A}=\frac{1}{2}\begin{bmatrix}1&1&0&0&0&0&0&0\\0&0&1&1&0&0&0&0\\0&0&0&0&1&1&0&0\\0&0&0&0&0&0&1&1\end{bmatrix} \tag{5}$$

and $\mathbf{A}^{t}$ is therefore the 8×4 horizontal averaging matrix.

From the equation (3) and (4), we can define the matrix $\mathbf{M}=\mathbf{AC}$ which performs IDCT and averaging at once as following equation:

$$\mathbf{x}'=\mathbf{ACXC}^{t}\mathbf{A}^{t}=\mathbf{MXM}^{t}. \tag{6}$$

$\mathbf{M}$ can be divided into two matrices by applying trigonometric identity $\cos\alpha+\cos\beta=2\cos[(\alpha+\beta)/2]\cos[(\alpha-\beta)/2]$ as $\mathbf{M}=\mathbf{C}_4^{'}\mathbf{D}$ where

$$\mathbf{C}_4^{'}=\begin{bmatrix}\frac{1}{\sqrt{2}}&\cos\frac{\pi}{8}&\cos\frac{2\pi}{8}&\cos\frac{3\pi}{8}&0&-\cos\frac{3\pi}{8}&-\cos\frac{2\pi}{8}&-\cos\frac{\pi}{8}\\\frac{1}{\sqrt{2}}&\cos\frac{3\pi}{8}&\cos\frac{6\pi}{8}&\cos\frac{9\pi}{8}&0&-\cos\frac{9\pi}{16}&-\cos\frac{6\pi}{16}&-\cos\frac{3\pi}{16}\\\frac{1}{\sqrt{2}}&\cos\frac{5\pi}{8}&\cos\frac{10\pi}{8}&\cos\frac{15\pi}{8}&0&-\cos\frac{15\pi}{16}&-\cos\frac{10\pi}{16}&-\cos\frac{5\pi}{16}\\\frac{1}{\sqrt{2}}&\cos\frac{7\pi}{8}&\cos\frac{14\pi}{8}&\cos\frac{21\pi}{8}&0&-\cos\frac{21\pi}{16}&-\cos\frac{14\pi}{16}&-\cos\frac{7\pi}{16}\end{bmatrix} \tag{7}$$

and

$$\mathbf{D}=\mathrm{diag}\left\{\frac{1}{2},\ \frac{1}{2}\cos\frac{\pi}{16},\ \frac{1}{2}\cos\frac{2\pi}{16},\ \frac{1}{2}\cos\frac{3\pi}{16},\ \frac{1}{2}\cos\frac{4\pi}{16},\ \frac{1}{2}\cos\frac{5\pi}{16},\ \frac{1}{2}\cos\frac{6\pi}{16},\ \frac{1}{2}\cos\frac{7\pi}{16}\right\}. \tag{8}$$

As $\mathbf{D}$ is a constant 8×8 diagonal matrix, it can be multiplied in advance at the dequantization stage. As we mentioned above, DQ stage multiplies each DCT coefficient by the quantizing step. We can perform just one multiplication with the quantizing step scaled with a diagonal term in $\mathbf{D}$, instead of two separate multiplications, first with a quantizing step and then with a diagonal term in $\mathbf{D}$. Therefore the multiplication of $\mathbf{D}$ can be absorbed in the dequantization stage without any increase in arithmetic complexity.

2.2 Rearranging IDCT

By rearranging operation sequence in IDCT, we can further reduce the number of arithmetic operations. As $\mathbf{C}_4'$ in (7) consists of two 4-point IDCT matrices $\mathbf{C}_4$ which are given by

$$\mathbf{C}_4 = \frac{1}{\sqrt{2}}\begin{bmatrix} \frac{1}{\sqrt{2}} & \cos\frac{\pi}{8} & \cos\frac{2\pi}{8} & \cos\frac{3\pi}{8} \\ \frac{1}{\sqrt{2}} & \cos\frac{3\pi}{8} & \cos\frac{6\pi}{8} & \cos\frac{9\pi}{8} \\ \frac{1}{\sqrt{2}} & \cos\frac{5\pi}{8} & \cos\frac{10\pi}{8} & \cos\frac{15\pi}{8} \\ \frac{1}{\sqrt{2}} & \cos\frac{7\pi}{8} & \cos\frac{14\pi}{8} & \cos\frac{21\pi}{8} \end{bmatrix}, \tag{9}$$

we can calculate $\mathbf{C}_4'$ using many well-known fast IDCT algorithms.[8,9] We adopted a fast IDCT algorithm based on the Winograd FFT algorithm.[10,11] This algorithm can be implemented like the flow graph of Figure 2(a), and represented in a matrix form as follows:

$$\mathbf{C}_4 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix}\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos\frac{\pi}{4} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & 1 \end{bmatrix}\begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2\sqrt{2}\cos\frac{\pi}{8}} & 0 & 0 \\ 0 & 0 & \frac{1}{2\sqrt{2}\cos\frac{2\pi}{8}} & 0 \\ 0 & 0 & 0 & \frac{1}{2\sqrt{2}\cos\frac{3\pi}{8}} \end{bmatrix}. \tag{10}$$

Unlike the other fast IDCT algorithms, it concentrates multiplications to the first stage of calculation. Using that algorithm, We can represent $\mathbf{C}_4'$ as the following equation and implement it as shown in Figure 2(b).

$$\mathbf{C}_4' = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix}\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos\frac{\pi}{4} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \end{bmatrix} \times$$
$$\operatorname{diag}\left\{\frac{1}{\sqrt{2}}, \frac{1}{2\cos\frac{\pi}{8}}, \frac{1}{2\cos\frac{2\pi}{8}}, \frac{1}{2\cos\frac{3\pi}{8}}, 0, \frac{1}{2\cos\frac{3\pi}{8}}, \frac{1}{2\cos\frac{2\pi}{8}}, \frac{1}{2\cos\frac{\pi}{8}}\right\}. \tag{11}$$

The last diagonal matrix in (11) can also be combined to the DQ stage just like the matrix $\mathbf{D}$. Consequently almost all of the multiplications in the combined IDCT-averaging stage are transferred to the DQ stage. All the operations we have to do

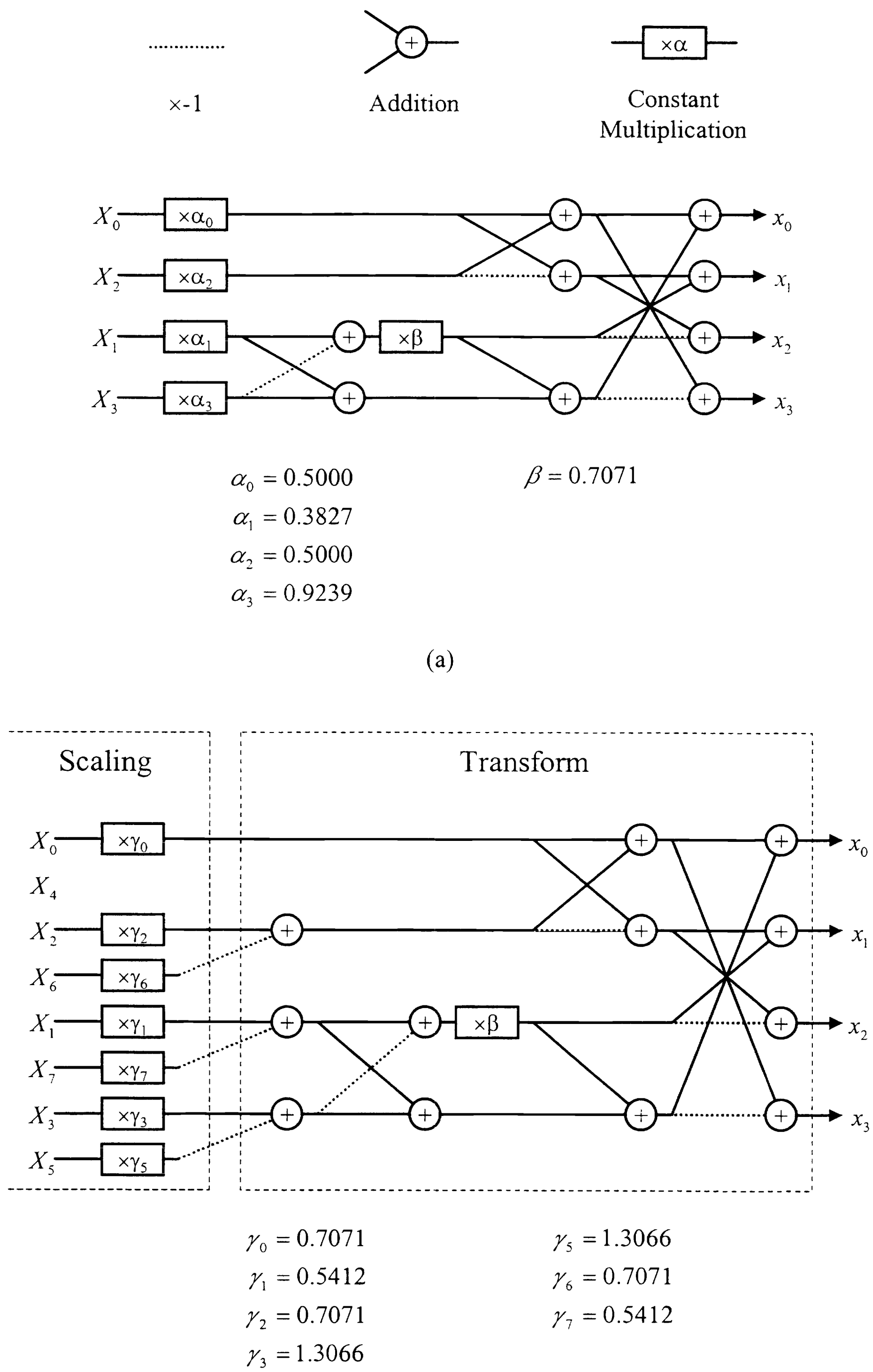

Figure 2: Flow graphs of (a) the 4×4 fast IDCT algorithm based on Winograd FFT and (b) the transform stage of the proposed down-conversion decoder.

$$\begin{bmatrix} q_{00} & q_{01} & q_{02} & q_{03} & q_{04} & q_{05} & q_{06} & q_{07} \\ q_{10} & q_{11} & q_{12} & q_{13} & q_{14} & q_{15} & q_{16} & q_{17} \\ q_{20} & q_{21} & q_{22} & q_{23} & q_{24} & q_{25} & q_{26} & q_{27} \\ q_{30} & q_{31} & q_{32} & q_{33} & q_{34} & q_{35} & q_{36} & q_{37} \\ q_{40} & q_{41} & q_{42} & q_{43} & q_{44} & q_{45} & q_{46} & q_{47} \\ q_{50} & q_{51} & q_{52} & q_{53} & q_{54} & q_{55} & q_{56} & q_{57} \\ q_{60} & q_{61} & q_{62} & q_{63} & q_{64} & q_{65} & q_{66} & q_{67} \\ q_{70} & q_{71} & q_{72} & q_{73} & q_{74} & q_{75} & q_{76} & q_{77} \end{bmatrix}$$

(a)

$$\frac{1}{8}\begin{bmatrix} 1.0000q_{00} & 0.7507q_{01} & 0.9239q_{02} & 1.5364q_{03} & 0 & 1.0266q_{05} & 0.3827q_{06} & 0.1493q_{07} \\ 0.7507q_{10} & 0.5635q_{11} & 0.6935q_{12} & 1.1533q_{13} & 0 & 0.7706q_{15} & 0.2873q_{16} & 0.1121q_{17} \\ 0.9239q_{20} & 0.6935q_{21} & 0.8536q_{22} & 1.4194q_{23} & 0 & 0.9484q_{25} & 0.3536q_{26} & 0.1379q_{27} \\ 1.5364q_{30} & 1.1533q_{31} & 1.4194q_{32} & 2.3604q_{33} & 0 & 1.5772q_{35} & 0.5879q_{36} & 0.2294q_{37} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.0266q_{50} & 0.7706q_{51} & 0.9484q_{52} & 1.5772q_{53} & 0 & 1.0538q_{55} & 0.3928q_{56} & 0.1533q_{57} \\ 0.3827q_{60} & 0.2873q_{61} & 0.3536q_{62} & 0.5879q_{63} & 0 & 0.3928q_{65} & 0.1464q_{66} & 0.0571q_{67} \\ 0.1493q_{70} & 0.1121q_{71} & 0.1379q_{72} & 0.2294q_{73} & 0 & 0.1533q_{75} & 0.0571q_{76} & 0.0223q_{77} \end{bmatrix}$$

(b)

Figure 3: The scaling matrices used in (a) the DQ stage of the conventional decoder, and (b) the scaling stage of the proposed decoder.

at the transform stage in Figure 1(b) are contained in the dotted box in Figure 2(b). We must apply the operations in Figure 2(b) in both vertical and horizontal direction at the stage so that we will obtain a 4×4 image block.

On the other hand, the scaling stage in Figure 1(b) does basically the same thing as the original DQ stage does. The number of operations is also exactly the same. What is different is that it uses a different scaling matrix. Let the matrix shown in Figure 3(a) be the quantization matrix used in the DQ stage. The scaling matrix used and saved in the scaling stage will then look like Figure 3(b). For instance, if the VLD stage sends a value of the position (1,2), the scaling stage multiplies it by $\frac{1}{8}\times 0.653q_{12}$, instead of q_{12}, to get $X(1,2)$. Note that $X(1,2)$ here means just an input datum to the transform stage, not a DCT coefficient.

4. EXPERIMENTAL RESULTS

We compared proposed down-conversion method with others, spatial domain down-conversion and 4×4 Cut.[3] Spatial domain scheme guarantees the minimal MSE and requires great number of operations. On the other hand, 4×4 Cut is the simplest DCT domain down-conversion. It requires only a few operations and produces some degradation. The other DCT domain down-conversion methods need at least twice as many operations as 4×4 Cut, while showing little improvement in image quality.

Table 1 compares the number of operations used. Because the DQ stage requires the same number of operations in all cases, we counted the operations in the transform stage only. By combining the averaging matrix with the IDCT matrix and using the fast IDCT algorithm, the total number of operations is reduced to 12 multiplications and 144 additions, while the direct calculation of 8×8 fast IDCT and spatial domain averaging requires 144 multiplications and 396 additions. The number of operations used in proposed method is comparable to 4×4 Cut, which requires 32 multiplications and 72 additions. The proposed method requires almost the same number of operations as 4×4 Cut, but the former guarantees optimal image quality while the latter doesn't.

We also compared the image quality of proposed method with existing schemes. We used eight 512×512 monochrome test images and compared the PSNR of down-converted image. The results are shown in Table 2. The proposed decimation scheme shows the same optimal quality as the spatial domain averaging and around 1dB higher PSNR than other DCT domain decimation methods.

Table 1: Comparison of the number of arithmetic operations

Number of operations	Proposed	Sptial domain down-conversion	4×4 Cut
Multiplication	12	144	32
Addition	144	396	72

Table 2: Comparison of the PSNR (dB)

Image	Proposed	Spatial domain down-conversion	4x4 Cut
baboon	54.38	54.38	53.46
bar	58.94	58.94	57.83
boats	69.44	69.44	68.58
bridge	59.12	59.13	58.28
couple	70.46	70.46	69.61
lax	57.54	57.54	56.64
lena	72.66	72.66	71.95
zelda	81.18	81.18	80.40
Average	65.47	65.47	64.59

5. CONCLUSIONS

In this paper we have proposed a down-conversion decoding scheme with the minimum MSE and the minimal number of arithmetic operations. We have derived an efficient algorithm in the DCT domain by first combining the DQ, IDCT, and spatial domain averaging stages and by concentrating most of the multiplications to the first stage. As shown in the experimental results, the proposed method has almost the same computational complexity as the simplest suboptimal method, while it shows the identical performance with the optimal spatial domain down-conversion. The method described in this paper can also be applied to 1:4 down-conversion.

REFERENCES

1. W. Kou and T. Fjalbrant, "A Direct Computation of DCT Coefficients for a Signal Block Taken from Two Adjacent Blocks," *IEEE Trans. Signal Processing* **39**, pp. 1692—1695, 1991.
2. S.-F. Chang and D. G. Messerschmitt, "Manipulation and Compositing of MC-DCT Compressed Video," *IEEE J. Selected Areas in Comm.* **13**, pp. 1-11, 1995.
3. S. Ng, "Low Resolution HDTV Receivers," *US Patent 5,262,854*, Nov. 1993
4. H. Sun, "Hierarchical Decoder for MPEG Compressed Video Data," *IEEE Trans. Consumer Electronics* **39**, pp. 559-564, 1993.
5. N. Merhav and V. Bhaskaran, "Fast Algorithms for DCT-Domain Image Down-Sampling and for Inverse Motion Compensation," *IEEE Trans. Circuits and Systems for Video Tech.* **7**, pp. 468-476, 1997.
6. K. K. Pang, H. G. Lim, S. Dunstan, J. M. Badcock, "Frequency Domain Decimation and Interpolation Techniques," *Picture Coding Symposium*, Melbourne, Australia, 1996.
7. A. Vetro, H. Sun, J. Bao, and T. Poon, "Frequency Domain Down-Conversion of HDTV Using Adaptive Motion Compensation," *Int. Conf. Image Processing*, pp. 763-766, Santa Barbara, CA, USA, 1997.
8. H. S. Hou, "A Fast Recursive Algorithm For Computing the Discrete Cosine Transform," *IEEE Trans. Acoust., Speech, Signal Processing* **35**, pp. 1455-1461, 1987.
9. S. C. Chan and K. L. Ho, "A New Two-Dimensional Fast Cosine Transform Algorithm", *IEEE Trans. Signal Processing*, **39**, pp. 481-485, 1991.
10. S. Winograd, "On Computing the Discrete Fourier Transform," *Mathematics of Computation* **32**, pp. 175-199, 1978.
11. Y. Arai, T. Agui, and M. Nakajima "A Fast DCT-SQ Scheme for Images," *Trans. IEICE* **E 71**, pp. 1095-1097, 1988.