IMAGE RESIZING IN AN ARBITRARY TRANSFORM DOMAIN

Hyungsuk Oh and Wonha Kim

College of Electronics and Information Engr. Kyunghee University
Yongin, Korea
E-mail: ogun5@khu.ac.kr , wonha@khu.ac.kr

Abstract

This paper develops a methodology for resizing image resolutions in an arbitrary block transform domain. To accomplish this, we represent the procedures resizing images in an arbitrary transform domain in the form of matrix multiplications from which the matrix scaling the image resolutions is produce. The experiments showed that the proposed method produces the reliable performances without increasing the computational complexity, compared to conventional methods when applied to various transforms.

Keywords: Image resizing, Block transform, H.264/AVC

1. Introduction

In the current integrated communication environment, resizing images or video frames according to the communication environment and the device being used is essential. Since image or video data to be compressed for the efficient transmission are expressed as transformed coefficients, it is extremely desirable to resize images directly in the transform domains [1]. As video or image coding technologies have evolved, various transforms such as the DCT, the modified DCT(MDCT) used in H.264/AVC and the Discrete Hadamard transform(DHT) have been adopted [2]. Therefore, there is an urgent need to develop an image resizing methodology applicable to an arbitrary transform domain. The current methods of resizing images in the transform domain have operated with only the DCT. Dugad et al. [3] removes the high frequency components and then they devised the matrix operations to resize the images. Park et al. [4] symmetrically extended the DCT block and then they exploited the convolution-multiplication theory to derive the window resizing image. Since the symmetric extension preserves the high frequency components of images, their method produces better image quality than that of Dugad et al. but increases more computational burden. Salazar et al. [5] and Shu etal. [6] developed techniques to scale images with arbitrary resolutions.

Unlike the DCT whose kernels are the real versions of the Fourier kernel, the kernels of the modified DCT (MDCT) are numbers approximating those of the DCT. So, it is not possible to express the MCDT kernels in mathematical formulation. The Hadamard transform is also not related to the DCT. This implies that the signal processing theories applicable to the DCT are not permissible to other transforms. Therefore, in general, the conventional methods developed in the DCT domain do not function in other transform domains.

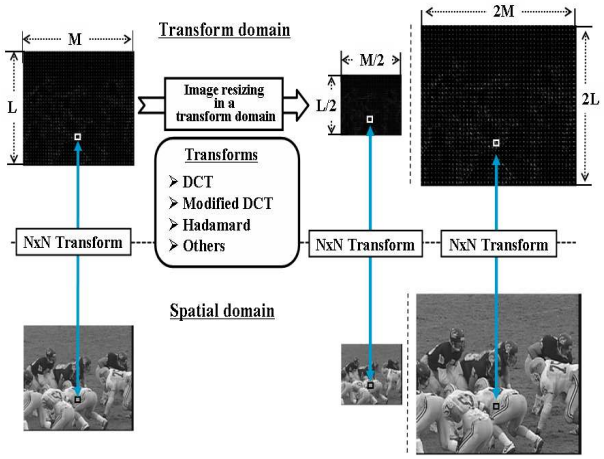

Fig 1. Overview of the developed image resizing methodology. The developed method can scale image resolution in an arbitrary block transform domain.

In this paper, we develop an image resizing methodology applicable to an arbitrary transform domain, as illustrated in Fig. 1. Our approach is to devise step-by-step procedures for resizing images in an arbitrary block transform domain and to represent each intermediate procedure in the form of matrix multiplications. These matrix multiplications produce a single scaling matrix that performs the entire procedure at once. Since all of the intermediate procedures are merged into the scaling matrix, the proposed method does not increase the computational complexity.

To confirm its applicability to an arbitrary image transform, we applied the proposed method to the DCT, the modified DCT (MDCT) used in in H.264/AVC [2] and the Discrete Hadamard transform (DHT). Although the performance varies according to the coding gain of the applied transform [7], the proposed method provides reliable performances for all of the tested transforms.

This paper is consisting of the following. In chapterⅡ, it introduces block transforms usually applied at image processing. In chapterⅢ, we develop the method for the down-sampling of image in an arbitrary block transform and In chapterⅣ, develop the method for the up-sampling. In chapterⅤ, we derive the matrices that perform the down/up scaling at the 8x8 DCT, the 4x4 modified DCT, the Hadamard Transform domain using the developed method. In chapterⅥ, the experiment shows that the proposed scheme is applied at an arbitrary transform domain and for a still image or a video. Finally, we say about the conclusion of this paper in chapterⅦ.

2. Orthogonal block transform

We refer to an $N\times N$ image block in the spatial domain and its corresponding transform domain block as $\mathbf{X}_{N\times N}$ and $\mathbf{X}_{N\times N}$, respectively. Let $\mathbf{T}_{N\times N}$ denote an NxN orthogonal block transform matrix. Then, $\mathbf{X}_{N\times N} = \mathbf{T}_{N\times N}\cdot\mathbf{x}_{N\times N}\cdot\mathbf{T}^{t}_{N\times N}$ and $\mathbf{x}_{N\times N} = \mathbf{T}^{t}_{N\times N}\cdot\mathbf{x}_{N\times N}\cdot\mathbf{T}_{N\times N}$. The transform matrices that are often used in image or video processing are as follows;

● Discrete Cosine Transform (DCT) [7]

$(k,n)^{th}\ N\times N$The element of the real DCT matrix is

$\mathbf{T}^{kn}_{NxN} = \mathbf{c}_k\sqrt{\frac{2}{N}}\cos\left[\left(\frac{\mathbf{n}+1}{2}\right)\frac{\mathbf{k}\pi}{\mathbf{N}}\right]$ for k,n=0,…,(N-1).

Where $\mathbf{c}_0 = \frac{1}{\sqrt{2}}$ for k=0 and $c_k = 1\ \ for\ k>0$.

● Modified DCT(MDCT) used in H.264/AVC [2][8]

The kernel of the MDCT approximates the DCT while it yields almost the same performance as the DCT. The 4x4 (N=4) MDCT matrix is

$$T_{4\times4} = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} \quad (1)$$

where $a=1/2, b=\sqrt{2/5}, c=\sqrt{1/10}$.

The 8x8 (N=8) MDCT matrix adopted by the H.264 extended profile is

$$T_{8\times8} = \begin{bmatrix} d & d & d & d & d & d & d & d \\ e & f & g & h & -h & -g & -f & -e \\ i & j & -j & -i & -i & -j & j & i \\ f & -h & -e & -g & g & e & h & -f \\ d & -d & -d & d & d & -d & -d & d \\ g & -e & h & f & -f & -h & e & -g \\ j & -i & i & -j & -j & i & -i & j \\ h & -g & f & -e & e & -f & g & -h \end{bmatrix} \quad (2)$$

where

$d=1/(2\sqrt{2}), e=12/(17\sqrt{2}), f=10/(17\sqrt{2}),$
$g=6/(17\sqrt{2}), h=3/(17\sqrt{2}), i=1/\sqrt{5}, j=1/(2\sqrt{5}).$

● Discrete Hadamard Transform [9]

$2^n\times2^n\ (N=2^n)$ The form of the Hadamard transform matrix is

$$\mathbf{T}_{2^n\times2^n} = \frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{T}_{2^{n-1}\times2^{n-1}} & \mathbf{T}_{2^{n-1}\times2^{n-1}} \\ \mathbf{T}_{2^{n-1}\times2^{n-1}} & -\mathbf{T}_{2^{n-1}\times2^{n-1}} \end{bmatrix}$$

where $\mathbf{T}_0 = [1]$.

3. Down-sampling in a transform domain

In this section, we develop a method of down-sampling an 2Nx2N transform domain block $\mathbf{X}_{2N\times2N} = \begin{bmatrix} \mathbf{X}^{0}_{N\times N} & \mathbf{X}^{1}_{N\times N} \\ \mathbf{X}^{2}_{N\times N} & \mathbf{X}^{3}_{N\times N} \end{bmatrix}$ to an NxN transform domain block $\mathbf{Y}_{N\times N}$. Also let $\mathbf{x}_{2N\times2N} = \begin{bmatrix} \mathbf{x}^{0}_{N\times N} & \mathbf{x}^{1}_{N\times N} \\ \mathbf{x}^{2}_{N\times N} & \mathbf{x}^{3}_{N\times N} \end{bmatrix}$, where $\mathbf{x}^{n}_{N\times N} = \mathbf{T}^{t}_{N\times N}\cdot\mathbf{X}^{n}_{N\times N}\cdot\mathbf{T}_{N\times N}$ for n=0, 1, 2, 3. Thus,

$$\mathbf{x}_{2N\times2N} = \begin{bmatrix} \mathbf{T}^{t}_{N\times N} & 0_{N\times N} \\ 0_{N\times N} & \mathbf{T}^{t}_{N\times N} \end{bmatrix}\cdot \begin{bmatrix} \mathbf{X}^{0}_{N\times N} & \mathbf{X}^{1}_{N\times N} \\ \mathbf{X}^{2}_{N\times N} & \mathbf{X}^{3}_{N\times N} \end{bmatrix}\cdot\begin{bmatrix} \mathbf{T}_{N\times N} & 0_{N\times N} \\ 0_{N\times N} & \mathbf{T}_{N\times N} \end{bmatrix}. \quad (3)$$

$\mathbf{Y}_{N\times N}$ is the transform domain block down-sampled from the spectrum of the image block $\mathbf{x}_{2N\times2N}$. Fig 2. illustrates the procedures of the developed down-sampling method. It should be noted that $\mathbf{X}_{2N\times2N} \neq \mathbf{T}_{2N\times2N}\cdot\mathbf{x}_{2N\times2N}\cdot\mathbf{T}^{t}_{2N\times2N}$ which implies that $\mathbf{X}_{2N\times2N}$ is not the spectrum of $\mathbf{x}_{2N\times2N}$. Denote $\mathbf{Y}_{2N\times2N}$ as the spectrum of $\mathbf{x}_{2N\times2N}$. Then,

$$\mathbf{Y}_{2N\times2N} = \mathbf{T}_{2N\times2N}\cdot\mathbf{x}_{2N\times2N}\cdot\mathbf{T}^{t}_{2N\times2N} \quad (4)$$

As shown in Fig. 1. the down-sampled image must take after the original image $\mathbf{X}_{2N\times2N}$ as much as possible. Thus, we should down-sample $\mathbf{Y}_{2N\times2N}$ that is the spectrum of $\mathbf{x}_{2N\times2N}$, instead of down-sampling $\mathbf{X}_{2N\times2N}$. Since the energy of an image signal is concentrated mostly in the low frequency range, the down-sampling filter should be a low-pass filter to preserve the low frequency components and remove the high frequency components. The matrix performing the low-pass filtering in a transform domain is $[\mathbf{I}_{N\times N}\quad 0_{N\times N}]$ where $\mathbf{I}_{N\times N}$ ans $0_{N\times N}$ are NxN identity matrix and zero matrix, respectively [3][4]. Therfore, the down-sampled transform domain block $\mathbf{Y}_{2N\times2N}$ is obtained such as;

$$\mathbf{Y}_{N\times N} = [\mathbf{I}_{N\times N}\quad 0_{N\times N}]\cdot\mathbf{Y}_{2N\times2N}\cdot[\mathbf{I}_{N\times N}\quad 0_{N\times N}]^{t} \quad (5)$$

By replacing $\mathbf{x}_{2N\times2N}$ of Eq. (4) by Eq. (3) and then plugging Eq. (4) into Eq. (5), we obtain;

$$\mathbf{Y}_{N\times N} = \left\{[\mathbf{I}_{N\times N}\quad 0_{N\times N}]\cdot\mathbf{T}_{2N\times2N}\cdot\begin{bmatrix} \mathbf{T}^{t}_{N\times N} & 0_{N\times N} \\ 0_{N\times N} & \mathbf{T}^{t}_{N\times N} \end{bmatrix}\right\}\cdot\begin{bmatrix} \mathbf{x}^{0}_{N\times N} & \mathbf{x}^{1}_{N\times N} \\ \mathbf{x}^{2}_{N\times N} & \mathbf{x}^{3}_{N\times N} \end{bmatrix}\cdot\left\{[\mathbf{I}_{N\times N}\quad 0_{N\times N}]\cdot\mathbf{T}_{2N\times2N}\cdot\begin{bmatrix} \mathbf{T}^{t}_{N\times N} & 0_{N\times N} \\ 0_{N\times N} & \mathbf{T}^{t}_{N\times N} \end{bmatrix}\right\}^{t} \quad (6)$$

where $\mathbf{D}_{N\times2N} = [\mathbf{I}_{N\times N}\quad 0_{N\times N}]\cdot\mathbf{T}_{2N\times2N}\cdot\begin{bmatrix} \mathbf{T}^{t}_{N\times N} & 0_{N\times N} \\ 0_{N\times N} & \mathbf{T}^{t}_{N\times N} \end{bmatrix}$.

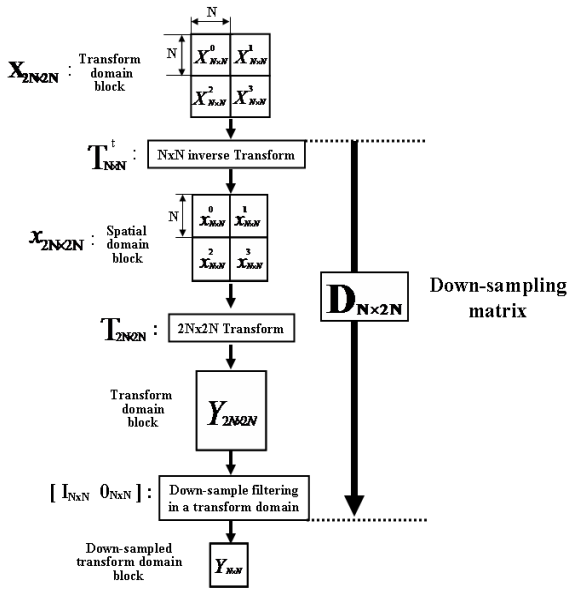

Fig 2. Down-sampling procedures. $\mathbf{X}_{2N\times 2N}$ is the input transform domain block. $\mathbf{Y}_{2N\times 2N}$ is the down-sampled transform domain block.

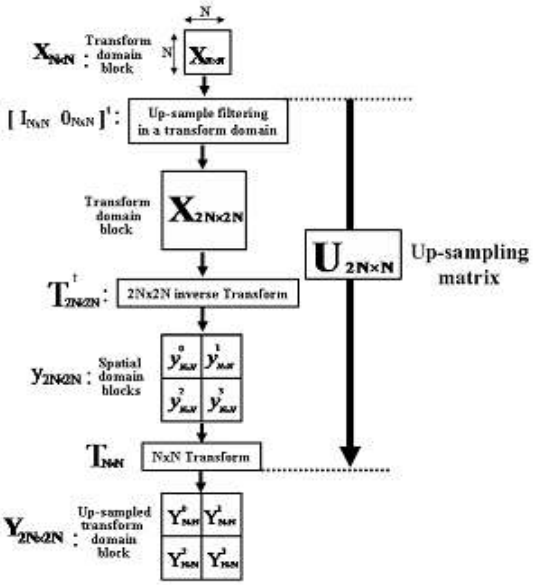

Fig 3. Up-sampling procedures. $\mathbf{X}_{N\times N}$ is the input transform domain block. $\mathbf{Y}_{2N\times 2N}$ is the up-sampled transform domain block.

$\mathbf{D}_{N\times 2N}$ is the matrix that performs down-sampling in a transform domain. $\mathbf{D}_{N\times 2N}$ is produced by the matrix multiplications representing each down- sampling procedure. Since the intermediate procedures eventually are merged into the matrix $\mathbf{D}_{N\times 2N}$, the proposed method does not increase the computational complexities compared to the conventional methods [4]. We will determine $\mathbf{D}_{N\times 2N}$ for various transforms in section V.

4. Up-sampling in a transform domain

In this section, we develop a method of up-sampling an NxN transform domain block $\mathbf{X}_{N\times N}$ to a 2Nx2N transform domain block $\mathbf{Y}_{2N\times 2N} = \begin{bmatrix} \mathbf{Y}^0_{N\times N} & \mathbf{Y}^1_{N\times N} \\ \mathbf{Y}^2_{N\times N} & \mathbf{Y}^3_{N\times N} \end{bmatrix}$. $\mathbf{Y}_{2N\times 2N}$ is not the spec- trum of the 2Nx2N image block up-sampled from the image block whose spectrum is $\mathbf{X}_{N\times N}$. Instead, each of the NxN blocks consisting of $\mathbf{Y}_{2N\times 2N}$ is the spectrum of each block at corresponding locations.

Fig. 3 illustrates the up-sampling procedures. $\mathbf{X}_{2N\times 2N}$ is obtained by up-sampling $\mathbf{X}_{N\times N}$. As seen in Fig. 1, the up-sampled image block whose spectrum is $\mathbf{X}_{2N\times 2N}$ must also resemble the image block of $\mathbf{X}_{N\times N}$. To fully exploit the low frequency components of image signals, the proper up-sampling filter matrix in a transform domain is $[\mathbf{I}_{N\times N} \quad 0_{N\times N}]^t$.

So,

$$\mathbf{X}_{2N\times 2N} = [\mathbf{I}_{N\times N} \quad 0_{N\times N}]^t \cdot \mathbf{X}_{N\times N} \cdot [\mathbf{I}_{N\times N} \quad 0_{N\times N}] \quad (7)$$

$\mathbf{X}_{2N\times 2N}$ is the spectrum of the 2Nx2N image block. However, the four NxN blocks constructing $\mathbf{X}_{2N\times 2N}$ are not the spectrum of each image block at the corresponding locations. Thus, $\mathbf{X}_{2N\times 2N}$ is not legitimate in the NxN transform domain. Let $\mathbf{y}_{2N\times 2N}$ be the 2Nx2N image block whose spectrum is $\mathbf{X}_{2N\times 2N}$. That is,

$$\begin{aligned} \mathbf{y}_{2N\times 2N} &= \mathbf{T}^t_{2N\times 2N} \cdot \mathbf{X}_{2N\times 2N} \cdot \mathbf{T}_{2N\times 2N} \\ &= \begin{bmatrix} \mathbf{y}^0_{N\times N} & \mathbf{y}^1_{N\times N} \\ \mathbf{y}^2_{N\times N} & \mathbf{y}^3_{N\times N} \end{bmatrix} \end{aligned} \quad (8)$$

The legitimate up-sampled transform domain block $\mathbf{Y}_{2N\times 2N}$ can be obtained by applying the NxN transform to the four NxN blocks of $\mathbf{Y}_{2N\times 2N}$. Therefore,

$$\begin{aligned} \mathbf{Y}_{2N\times 2N} = &\begin{bmatrix} \mathbf{T}_{N\times N} & 0_{N\times N} \\ 0_{N\times N} & \mathbf{T}_{N\times N} \end{bmatrix} \cdot \\ &\begin{bmatrix} \mathbf{X}^0_{N\times N} & \mathbf{X}^1_{N\times N} \\ \mathbf{X}^2_{N\times N} & \mathbf{X}^3_{N\times N} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{T}_{N\times N} & 0_{N\times N} \\ 0_{N\times N} & \mathbf{T}_{N\times N} \end{bmatrix}. \end{aligned} \quad (9)$$

By replacing $\mathbf{X}_{2N\times 2N}$ of Eq. (8) by Eq. (7) and then plugging Eq. (8) into Eq. (9), we obtain

$$\mathbf{Y}_{2N\times 2N} = \mathbf{U}_{2N\times N} \cdot \mathbf{X}_{N\times N} \cdot \mathbf{U}^t_{2N\times N} \quad (10)$$

$\mathbf{U}_{2N\times N}$ is also the matrix that performs up-sampling in a transform domain.

5. Up/Down-sampling matrices

In this section, we illustrate the up/down-sampling matrices for various transforms. From Eq. (6) and Eq. (10), it can be known that $\mathbf{D}_{\mathbf{N\times 2N}} = \mathbf{U}^{\mathbf{t}}_{\mathbf{2N\times N}}$.

- 8x8 DCT

By plugging the 8x8 DCT and the 16x16 DCT matrices into $\mathbf{T}_{8\times 8}$ and $\mathbf{T}_{16\times 16}$ of Eq. (6), the matrix performing down-sampling in the 8x8 DCT domain is obtained as in Eq. (11). Then, the up-sampling matrix is $\mathbf{U}^{\mathbf{DCT}}_{16\times 8} = \left(\mathbf{D}^{\mathbf{DCT}}_{16\times 8}\right)^{\mathbf{t}}$

$$\mathbf{D}^{\mathbf{DCT}}_{16\times 8} = \left[\mathbf{I}_{8\times 8} \quad 0_{8\times 8}\right]\cdot \mathbf{T}_{16\times 16}\cdot \begin{bmatrix} \mathbf{T}^{\mathbf{t}}_{8\times 8} & 0_{8\times 8} \\ 0_{8\times 8} & \mathbf{T}^{\mathbf{t}}_{8\times 8} \end{bmatrix} \tag{11}$$

$$= \begin{bmatrix} 0.7071 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.6376 & 0.2986 & -0.0585 & 0.0241 & -0.0125 & 0.0071 & -0.0039 & 0.0018 \\ 0 & 0.7071 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.2163 & 0.5446 & 0.3812 & -0.0951 & 0.0436 & -0.0235 & 0.0128 & -0.0057 \\ 0 & 0 & 0.7071 & 0 & 0 & 0 & 0 & 0 \\ 0.1326 & -0.2219 & 0.5081 & 0.4008 & -0.1061 & 0.0493 & -0.0253 & 0.0110 \\ 0 & 0 & 0 & 0.7071 & 0 & 0 & 0 & 0 \\ -0.0985 & 0.1509 & -0.2024 & 0.4971 & 0.4065 & -0.1078 & 0.0476 & -0.0196 \end{bmatrix}$$

$$\begin{bmatrix} 0.7071 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.6376 & 0.2986 & 0.0585 & 0.0241 & 0.0125 & 0.0071 & 0.0039 & 0.0018 \\ 0 & -0.7071 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.2163 & 0.5446 & -0.3812 & -0.0951 & -0.0436 & -0.0235 & -0.0128 & -0.0057 \\ 0 & 0 & 0.7071 & 0 & 0 & 0 & 0 & 0 \\ -0.1326 & -0.2219 & -0.5081 & 0.4008 & 0.1061 & 0.0493 & 0.0253 & 0.0110 \\ 0 & 0 & 0 & 0.7071 & 0 & 0 & 0 & 0 \\ 0.0985 & 0.1509 & 0.2024 & 0.4971 & -0.4065 & -0.1078 & -0.0476 & -0.0196 \end{bmatrix}$$

- 4x4 modified DCT(MDCT) used in H.264/AVC

By plugging the 4x4 MDCT and the 8x8 MDCT matrices into $\mathbf{T}_{4\times 4}$ and $\mathbf{T}_{8\times 8}$ of Eq. (6), the matrix performing down-sampling in the 4x4 MDCT domain such as

$$\mathbf{D}^{\mathbf{MDCT}}_{4\times 8} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ \frac{31}{34\sqrt{2}} & \frac{11}{17\sqrt{5}} & -\frac{1}{34\sqrt{2}} & \frac{1}{34\sqrt{5}} & -\frac{31}{34\sqrt{2}} & \frac{11}{17\sqrt{5}} & \frac{1}{34\sqrt{2}} & \frac{1}{34\sqrt{5}} \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 \\ -\frac{11}{34\sqrt{2}} & \frac{41}{34\sqrt{5}} & \frac{19}{34\sqrt{2}} & -\frac{1}{17\sqrt{5}} & \frac{11}{34\sqrt{2}} & \frac{41}{34\sqrt{5}} & -\frac{19}{34\sqrt{2}} & -\frac{1}{17\sqrt{5}} \end{bmatrix}$$

- 4x4 Discrete Hadamard transform (DHT)

In the same way, the down-sampling matrix in the 4x4 DHT domain is

$$\mathbf{D}^{\mathbf{DHT}}_{4\times 8} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 \end{bmatrix}$$

6. Experiment and Discussion

To verify that it performs image resizing in an arbitrary transform domain, we apply the proposed method to the 8×8 DCT, the 4×4 modified DCT (MDCT) used in H.264/AVC and the 4×4 DHT [2][9]. The original images transformed by each transform were first down-sampled and up-sampled again by each method. We use the down-sampling matrix $\mathbf{D}_{\mathbf{N\times 2N}}$ and the up-sampling matrix $\mathbf{U}_{\mathbf{2N\times N}}$ which are obtained in section V. The size of the test images is CIF (352×288). The degradation between the down/up-sampled image and the original image was evaluated by the PSNR. Table 1 compares the PSNR for the various transforms. Fig. 4 shows the restored images and the difference images between the restored and original images.

The 8×8 DCT produces the best performance, since its transform coding gain is the highest among the tested transforms[7]. Particularly, in the DCT domain, the up/down-sampling matrices obtained by the proposed method are the same as the scaling windows derived by Park's method whose performance is rated amongst the best [4]. It can also be observed that the proposed method and Park's method produce even better results than Dugad's method [3], since they better preserve the high frequency characteristics of the images. Therefore, it is confirmed that the proposed method operates in the DCT domain.

Since the size of the MDCT of H.264/AVC is 4x4, it is reasonable to compare the performances of the methods using 4x4 transforms. The PSNR of 4×4 MDCT and the 4×4 DCT using Park's method are within 0.15 dB of each other and the subjective restored image qualities are indistinguishable. Considering that the transform coding gain of MDCT is slightly lower than that of DCT [2]. we can judge that the performances of 4×4 MDCT are reliable. Therefore, the proposed method can be also applicable to the 4×4 MDCT.

The relatively low PSNR of the DHT is due to its low transform coding gain compared to those of the other transforms. The performances of the proposed method in the DHT domain are almost same as those of the bi-linear interpolation in the spatial domain. This is because the DHT kernels are generated from the Haar basis, which is equivalent to the bilinear filter [9]. This implies that the proposed method also provides reasonable performance in the DHT domain.

We also compared the overall performances of the image resizing methods in transform domains and the interpolation methods in the spatial domain. The tested spatial interpolation filters are the bi-linear filter and the filter used in MPEG [10]. As shown in Table 1, the methods in transform domain show better results because they compactly remove the high frequency components where the energy of the image signals rarely exists, while maintain well the low components where most of the energy lies.

When applied to the DCT and the MDCT, the proposed method requires 4.2 multiplications and 6.7 additions per pixel, which is same as that in the case of the Park's method [4]. In the DHT, the proposed method requires 2 multiplications and 2 additions per pixel. Thus, the proposed method does not increase the complexity compared to the conventional methods.

Table 1. PSNR (dB) comparisons of original images and up/down-sampled images for each transform domain .

	Transform Domain			Spatial Domain	
	8x8 DCT	4x4 MDCT	4x4 DHT	Bi-linear	MPEG
Foreman	33.21	32.58	29.85	29.85	31.33
Bus	27.69	27.14	25.67	25.67	26.02
Football	28.47	27.62	25.63	25.63	26.60
Harbour	34.13	32.00	28.06	28.06	31.34
Crew	38.17	37.05	34.61	34.61	36.54
Soccer	31.60	30.94	29.48	29.48	30.10
City	30.33	29.37	28.28	28.28	28.04

7. Conclusion

This paper develops a method of changing the image resolutions in an arbitrary block transform domain, while the conventional methods can operate only in the DCT domain. The experiments confirmed that the proposed method is able to scale the image resolution in various transform domains, while providing reliable performances. Therefore, it can be concluded that the proposed method can scale the image resolution in an linear block transform domains. In a future study, the proposed method should be improved to enable it to handle arbitrary images sizes.

Acknowledgement
This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD). (KRF-2007-511-20080082)

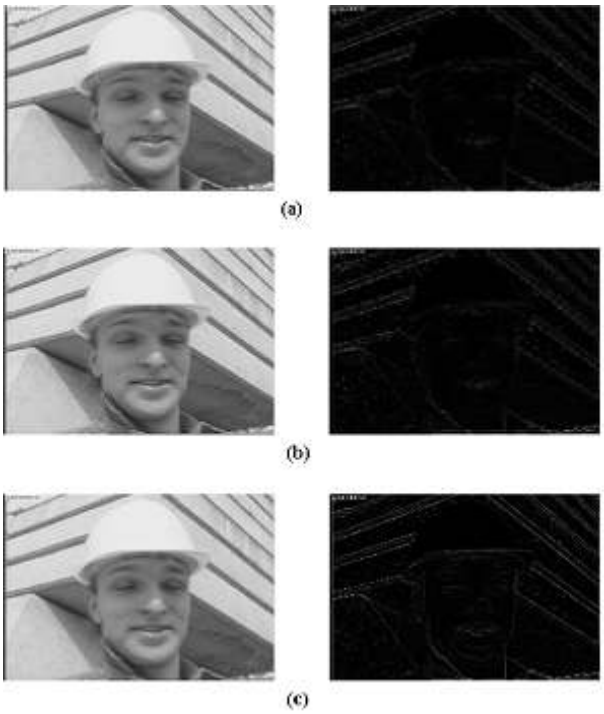

Fig 4. Down-sampled and up-sampled images at various transform domains and their difference images from the original images. The image size is CIF(352x288). (a) 8×8 DCT domain, (b) 4×4 Modified DCT domain of H.264/AVC and (c) 4×4 Discrete Hadamard Tranform domain.

References

[1] H. Sun, X. Chen, and T. Chiang, "Digital Video Transcoding for Transmission and Storage," CRC Press, 2005.

[2] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in h.264/avc," IEEE Trans. Circ. and Syst. Video Tech., vol. 13, no. 7, pp. 598.603, July 2003.

[3] R. Dugad and N. Ahuja, "A fast scheme for image size change in the compressed domain," IEEE Trans. Circ. and Syst. Video Tech., vol. 11, no. 4, pp. 461.474, Apr. 2001.

[4] H.W. Park, Y.S. Park, and S.K.O, "L/m-fold image resizing in block-dct domain using symmetric convolution," IEEE Trans. Image Processing, vol. 12, no. 9, pp. 1016.1034, Sept. 2003.

[5] C.L. Salazar and T.D. Tran, "A complexity scalable universal dct domain image resizing algorithm," IEEE Trans. Circ. and Syst. Video Tech., vol. 17, no. 4, pp. 495.499, Apr. 2007.

[6] H. Shu and L.P. Chau, "An efficient arbitrary downsizing algorithm for video transcoding," IEEE Trans. Circ. and Syst. Video Tech., vol. 14, no. 6, pp. 887.891, June 2004.

[7] G. Strang and T.Q. Nguyen, "Wavelets and Filter Banks," Wellesley-Cambridge Press, 1997.

[8] S. Gorden, "Simplified use of 8x8 transforms," JVT-I022, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG 9th Meeting : San Diego USA, Sept. 2003.

[9] T.K. Moon and W.C. Stirling, "Mathematical Methods and Algorithms for Signal Processing," Prentice-Hall, 2000.

[10] J. Vieron, M. Wien, and H. Schwarz, "Jsvm 11 software," JVT-X203, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG 24th Meeting : Geneva Switzerland, June 2007.