# Arbitrary-Ratio Image Resizing Using Fast DCT of Composite Length for DCT-Based Transcoder

Young Seo Park and Hyun Wook Park, *Senior Member, IEEE*

***Abstract*—We propose a fast arbitrary-ratio image resizing method for transcoding of the compressed images. The downsizing process in the discrete cosine transform (DCT) domain can be implemented by truncating high-frequency coefficients, whereas the upsizing process is implemented in the DCT domain by padding zero coefficients to the high-frequency part. The proposed method combines a fast inverse and forward DCT of composite length for arbitrary-ratio upsizing or downsizing. According to the resizing ratio, truncating the high-frequency coefficients and padding zeros are appropriately considered by combining the inverse DCT and forward DCT. The proposed method shows a good peak signal-to-noise ratio and less computational complexity compared with the spatial-domain and previous DCT-domain image resizing methods.**

***Index Terms*—Arbitrary-ratio image resizing, composite length DCT, transcoder.**

## I. Introduction

With advances in digital signal processing and digital media, many video streams and images are now transmitted and stored in compressed forms. Accordingly, image manipulation in a compressed domain is usually more efficient than that in a spatial domain in terms of computation complexity and image quality. Numerous approaches have been applied to image and video processing in the compressed domain. Among the many image manipulation processes, image resizing is a fundamental and important operation for various multimedia applications. Previous studies have demonstrated that image resizing in the compressed domain rather than in the spatial domain is generally efficient in terms of image quality and computation amount [1]–[4].

Straightforward image resizing of a compressed image can be performed by a sequential operation such as decompression of a compressed image, resizing of the decompressed image, and recompression of the resized image. However, these approaches are undesirable owing to high computational cost. Many approaches for image resizing have also been developed in the transform domain. A downsizing process in the discrete cosine transform (DCT) domain can be implemented by truncating high-frequency DCT coefficients, whereas an upsizing process is implemented in the transform domain by padding zero coefficients to the high-frequency side. Recent approaches for image resizing in the transform domain yielded good results in both computational complexity and image quality [1], [2]. Image resizing methods have been performed in the DCT domain by a low-pass truncated approximation [1] and a subband approximation [2], respectively, both of which show good peak signal-to-noise ratio (PSNR) performance.

Some approaches for image resizing in the DCT domain showed good results in both computational complexity and image quality. However, these methods are limited with respect to arbitrary-ratio image resizing [1], [2]. Other recent approaches proposed arbitrary-ratio image resizing methods [3], [4]. Shu and Chau's method reconstructs $8 \times 8$ an resized block from neighbor input blocks and their shift matrices in the DCT domain, which requires relatively high computational complexity. Moreover, if $(M/L) \times 8$ is not an integer value when the image is downsized with a ratio of $L/M(M > L)$, this ratio could not be implemented by Shu and Chau's method. Our previous method also has good image quality for arbitrary-ratio image resizing [4]. The basic concept of $L/M$-fold resizing is implemented by performing $M$-fold downsizing after $L$-fold upsizing. However, the computational complexity of this method is not efficient when $L$ and $M$ are large values. Salazar and Tran proposed a generalized method for arbitrary-ratio image resizing recently [5]. However, it did not suggest an optimum length of DCT and inverse DCT (IDCT) for high PSNR.

We propose a fast arbitrary-ratio image resizing method in the DCT domain, which produces visually fine images with high PSNR. Furthermore, the proposed method is faster than the previous arbitrary-ratio resizing methods. The arbitrary-ratio image resizing can be realized by the fast inverse and forward DCT of composite length [6], [7]. We analyze the computational complexity and the PSNR, and then compare them with those of previous methods.

In the paper, a new image resizing method that can reduce computational complexity and improve the PSNR is described. In Section II, the proposed image resizing method is outlined. Section III describes another proposed approach for faster resizing, which further reduces computation complexity with a cost of slight PSNR degradation. Experimental results are given in Section IV while Section V presents conclusions.

## II. Proposed Image Resizing Method in the DCT Domain (Case I)

The proposed fast arbitrary-ratio image resizing method is performed by combining the IDCT and forward DCT (DCT) of

Manuscript received August 19, 2004; revised March 3, 2005. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Luca Lucchese.

Y. S. Park was with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 305-701, Korea. He is now with Ontimetek, Sungnam-si, Kyunggi-do, Korea (e-mail: young@athena.kaist.ac.kr).

H. W. Park is with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 305-701, Korea (e-mail: hwpark@ee.kaist.ac.kr).

Digital Object Identifier 10.1109/TIP.2005.863117

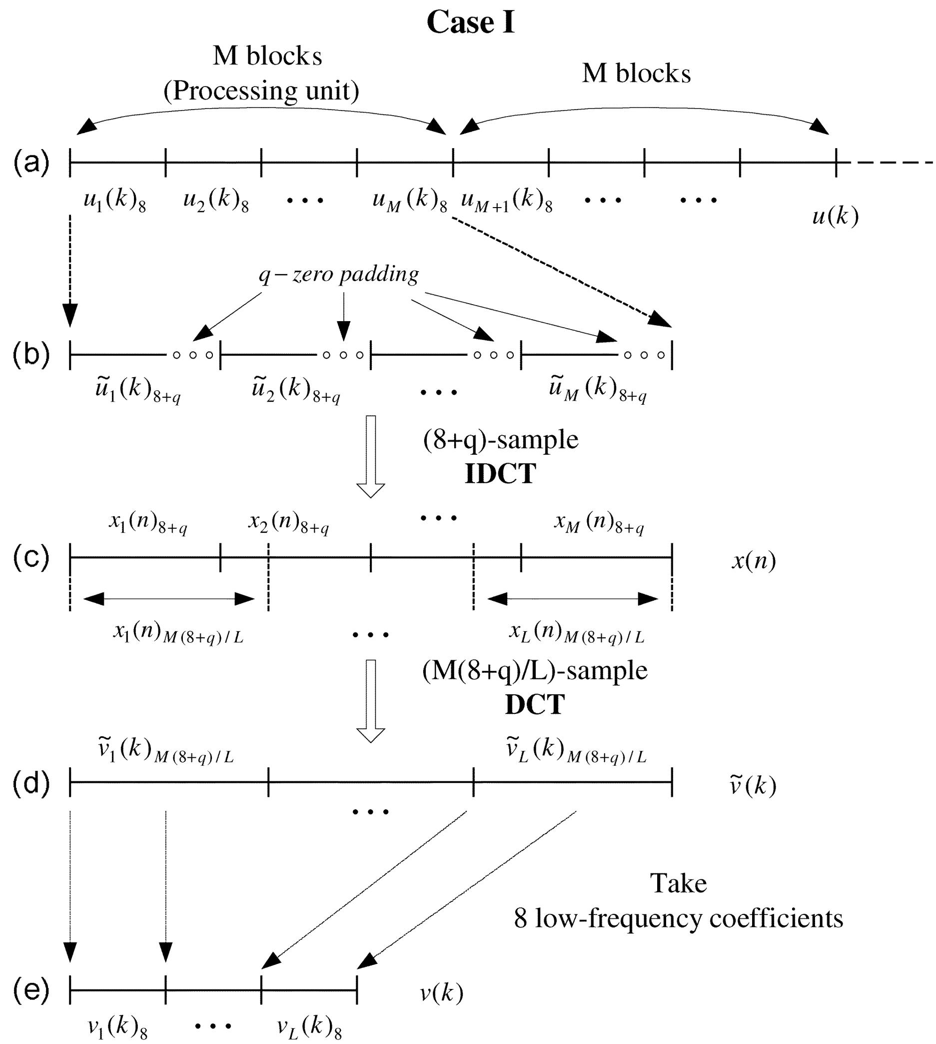


Fig. 1. Conceptual diagram of the proposed $L/M$-fold resizing (Case I): (a) processing unit with $M$ blocks, (b) zero padding in the DCT domain, (c) $(8+q)$-sample IDCT for $M$ blocks and splitting $M$ blocks into $L$ blocks in spatial domain, (d) $(M(8+q)/L)$-sample DCT for $L$ blocks, and (e) resizing by truncating high-frequency parts in the DCT domain.

composite length sequentially. In general, a downsized image in the DCT domain can be obtained by truncating the high-frequency DCT coefficients and an upsized image is implemented by zero padding. According to the resizing ratio, the number of truncated coefficients and the number of padded zeros are appropriately determined, and the IDCT and DCT of the corresponding lengths are performed.

The proposed $L/M$-fold resizing is performed by the sequential operations of the IDCT and DCT. For a two-dimensional (2-D) image, the one-dimensional (1-D) sequential operations of the IDCT and DCT are performed in the horizontal direction and then in the vertical direction. Therefore, different resizing ratios can be applied to the horizontal and vertical directions, respectively. This paper describes the 1-D resizing method, which can be applied to a 2-D image independently in the horizontal and vertical directions.

Let $L/M$ be an irreducible fractional number. The proposed method of $L/M$-fold resizing is processed with $M$-blocks unit, each block having a length of eight. First, the $(8+q)$-sample IDCT is performed for $M$-blocks. On this occasion, zero padding of $q$ samples is required in advance. The inverse transformed sequences have a total $(8+q) \times M$ of samples in a group of $M$ blocks, which are retransformed by DCT with a length of $(8+q) \times (L/M)$. We then truncate the high-frequency coefficients except eight low-frequency coefficients, thereby reconstructing $L$ blocks of eight samples in the DCT domain. The resized image can be obtained through this IDCT and DCT processing. No further processing is needed.

A conceptual diagram of the $L/M$-fold resizing is shown in Fig. 1. For $L/M$-fold resizing, $M$ consecutive 8-sample blocks are converted to $L$ 8-sample blocks. Therefore, $M$ 8-sample blocks are grouped as shown in Fig. 1(a). $u_i(k)_8 (i = 1, \ldots, M)$denotes 8-sample DCT coefficients of the $i$th block in a group of $M$ blocks. The total sample number of the original $M$ blocks is $8 \times M$. In order to perform the proposed $L/M$-fold resizing, the proposed method requires that the number of total samples in a group of $M$blocks should be a common multiple of $L$ and $M$. If not, zero padding is required for each block. Constraints of the proposed resizing method are given as follows:

$$\tilde{N} = \text{common multiple of } \{M, L\} \quad : \text{Constraint I}$$
$$\tilde{N} \geq \text{Max}\{M, L\} \times 8 \quad : \text{Constraint II} \qquad (1)$$

where $\tilde{N}$is the required number of total samples in a group of $M$ blocks, which is determined as follows:

$$\tilde{N} = M \times (8+q), \quad q \geq 0 \tag{2}$$

where $q$ is the smallest positive integer value satisfying the constraints in (1). Consequently, $q$is the required zero padding number per block for Case I.

Fig. 1(b) shows a case when $8 \times M(=N)$ is not the common multiple of $L$ and $M$. An IDCT of length of $(8+q)$ is performed for the zero-padded blocks. $x_i(n)_{8+q}$ represents $(8+q)$-sample sequence of the $i$th block in the spatial domain, and $\tilde{u}_i(k)_{8+q}$ are the $(8+q)$-sample DCT coefficients of $x_i(n)_{8+q}$. Let $\mathbf{x^i_{8+q}}$ be the vector form of $x_i(n)_{\mathbf{8+q}}$ and $\mathbf{\tilde{u}^i_{8+q}}$ be the vector of $\tilde{u}_i(k)_{8+q}$. Then, a vector of group of $M$ blocks can be described as follows:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x^1_{8+q}} \\ \mathbf{x^2_{8+q}} \\ \vdots \\ \mathbf{x^M_{8+q}} \end{bmatrix} = \begin{bmatrix} \mathbf{C^{-1}_{8+q}} & \mathbf{0_{8+q}} & \mathbf{0_{8+q}} & \mathbf{0_{8+q}} \\ \mathbf{0_{8+q}} & \mathbf{C^{-1}_{8+q}} & \mathbf{0_{8+q}} & \mathbf{0_{8+q}} \\ \mathbf{0_{8+q}} & \mathbf{0_{8+q}} & \ddots & \mathbf{0_{8+q}} \\ \mathbf{0_{8+q}} & \mathbf{0_{8+q}} & \mathbf{0_{8+q}} & \mathbf{C^{-1}_{8+q}} \end{bmatrix} \begin{bmatrix} \mathbf{\tilde{u}^1_{8+q}} \\ \mathbf{\tilde{u}^2_{8+q}} \\ \vdots \\ \mathbf{\tilde{u}^M_{8+q}} \end{bmatrix} \tag{3}$$

where $\mathbf{C^{-1}_{8+q}}$ is the $(8+q)$-sample IDCT matrix and $\mathbf{0_{8+q}}$ is the $\mathbf{8+q \times 8+q}$ zero matrix.

Since the number of total samples in a group of $M$ blocks becomes a multiple of $L$ by appending $q$ zeros to each DCT block, $\mathbf{x}$ can be split into $L$-blocks as shown in Fig. 1(c). It can be described as follows:

$$\begin{bmatrix} \mathbf{x^1_{M(8+q)/L}} \\ \mathbf{x^2_{M(8+q)/L}} \\ \cdots \\ \mathbf{x^L_{M(8+q)/L}} \end{bmatrix} = \mathbf{x}. \tag{4}$$

Then the $(M(8+q)/L)$-sample DCT is performed for $L$ blocks in (4). We can then extract eight DCT coefficients of low frequency from $(M(8+q)/L)$ coefficients in each block. Finally, the resized sequence $\mathbf{v^i_8}(i=1,\ldots,L)$ can be obtained as follows:

$$\begin{bmatrix} \mathbf{v^1_8} \\ \mathbf{v^2_8} \\ \vdots \\ \mathbf{v^L_8} \end{bmatrix} = \begin{bmatrix} \mathbf{C^{U8}_{M(8+q)/L}} & \mathbf{0_{M(8+q)/L}} & \mathbf{0_{M(8+q)/L}} & \mathbf{0_{M(8+q)/L}} \\ \mathbf{0_{M(8+q)/L}} & \mathbf{C^{U8}_{M(8+q)/L}} & \mathbf{0_{M(8+q)/L}} & \mathbf{0_{M(8+q)/L}} \\ \mathbf{0_{M(8+q)/L}} & \mathbf{0_{M(8+q)/L}} & \ddots & \mathbf{0_{M(8+q)/L}} \\ \mathbf{0_{M(8+q)/L}} & \mathbf{0_{M(8+q)/L}} & \mathbf{0_{M(8+q)/L}} & \mathbf{C^{U8}_{M(8+q)/L}} \end{bmatrix} \times \begin{bmatrix} \mathbf{x^1_{M(8+q)/L}} \\ \mathbf{x^2_{M(8+q)/L}} \\ \cdots \\ \mathbf{x^L_{M(8+q)/L}} \end{bmatrix} = \mathbf{x} \tag{5}$$

where $\mathbf{C^{U8}_{N(8+q)/L}}$ is the $8 \times (M(8+q)/L)$ matrix of the upper 8 rows of $(M(8+q)/L)$-sample DCT matrix and $\mathbf{0_{M(8+q)/L}}$ is the $8 \times (M(8+q)/L)$ zero matrix.

Consequently, the resized DCT coefficients $\mathbf{v^i_8}(i=1,\ldots,L)$ are obtained by applying (3) and (5) as follows:

$$\begin{bmatrix} \mathbf{v^1_8} \\ \mathbf{v^2_8} \\ \vdots \\ \mathbf{v^L_8} \end{bmatrix} = \begin{bmatrix} \mathbf{C^{U8}_{M(8+q)/L}} & \mathbf{0_{M(8+q)/L}} & \mathbf{0_{M(8+q)/L}} & \mathbf{0_{M(8+q)/L}} \\ \mathbf{0_{M(8+q)/L}} & \mathbf{C^{U8}_{M(8+q)/L}} & \mathbf{0_{M(8+q)/L}} & \mathbf{0_{M(8+q)/L}} \\ \mathbf{0_{M(8+q)/L}} & \mathbf{0_{M(8+q)/L}} & \ddots & \mathbf{0_{M(8+q)/L}} \\ \mathbf{0_{M(8+q)/L}} & \mathbf{0_{M(8+q)/L}} & \mathbf{0_{M(8+q)/L}} & \mathbf{C^{U8}_{M(8+q)/L}} \end{bmatrix} \times \begin{bmatrix} \mathbf{C^{-1}_{8+q}} & \mathbf{0_{8+q}} & \mathbf{0_{8+q}} & \mathbf{0_{8+q}} \\ \mathbf{0_{8+q}} & \mathbf{C^{-1}_{8+q}} & \mathbf{0_{8+q}} & \mathbf{0_{8+q}} \\ \mathbf{0_{8+q}} & \mathbf{0_{8+q}} & \ddots & \mathbf{0_8+q} \\ \mathbf{0_{8+q}} & \mathbf{0_{8+q}} & \mathbf{0_{8+q}} & \mathbf{C^{-1}_{8+q}} \end{bmatrix} \begin{bmatrix} \mathbf{\tilde{u}^1_{8+q}} \\ \mathbf{\tilde{u}^2_{8+q}} \\ \vdots \\ \mathbf{\tilde{u}^M_{8+q}} \end{bmatrix}. \tag{6}$$

In the resizing process, we need a scaling operation to compensate the effects of truncation and zero padding of DCT coefficients. Let us take $J$ samples from eight DCT coefficients, i.e., $(J-8)$ zeros are padded when $J > 8$ and $(8-J)$ samples are truncated when $J < 8$. If we take $J$-sample IDCT of the above $J$ DCT coefficients, the transformed data should be scaled with $(\sqrt{J})/(\sqrt{8})$, since the DCT coefficients were originally obtained by 8-sample DCT. In a similar way, if we perform $K$-sample DCT and take eight DCT coefficients from $K$ DCT coefficients by truncation or zero padding, the eight DCT coefficients should be scaled with $(\sqrt{8})/(\sqrt{K})$. Consequently, if we take $J$-sample IDCT and $K$-sample DCT for $L/M$-fold resizing in 1-D, the resized DCT coefficients should be scaled with $(\sqrt{J})/(\sqrt{K})$.

We briefly show an example of $3/4$-fold downsizing using our method. First, each group consists of four blocks. The total number of samples in a group is 32. Since the number 32 is not a multiple of 3, one zero padding into each block $(q=1)$ is required to satisfy constraints I and II in (1). Then, each block is transformed by a 9-sample IDCT and retransformed by a 12-sample DCT. Eight coefficients of low frequency are taken among 12 DCT coefficients in each of $L$ blocks.

We adopt Bi and Yu's algorithm for fast DCT of composite length [7]. The IDCT of length 9 required eight multiplications and 34 additions, which was optimized by Bi and Yu. And the DCT of length 12 required 14 multiplications and 49 additions. The detailed computation amount of the proposed method is shown in the experimental results, which is based on Bi and Yu's analysis of the composite-length DCT [7].

In the case of $4/3$-fold upsizing, the upsizing is processed with a group of 3-blocks. The total number of samples in a group is 24. This is also a multiple of four. However, the number of 24 does not satisfy constraint II in (1). Thus, we should select $\tilde{N} = 36$ instead of 24 in order to satisfy constraints I and II in (1). This means four zeros $(q=4)$ are appended to each block. Then, 12-sample inverse DCTs for three blocks and

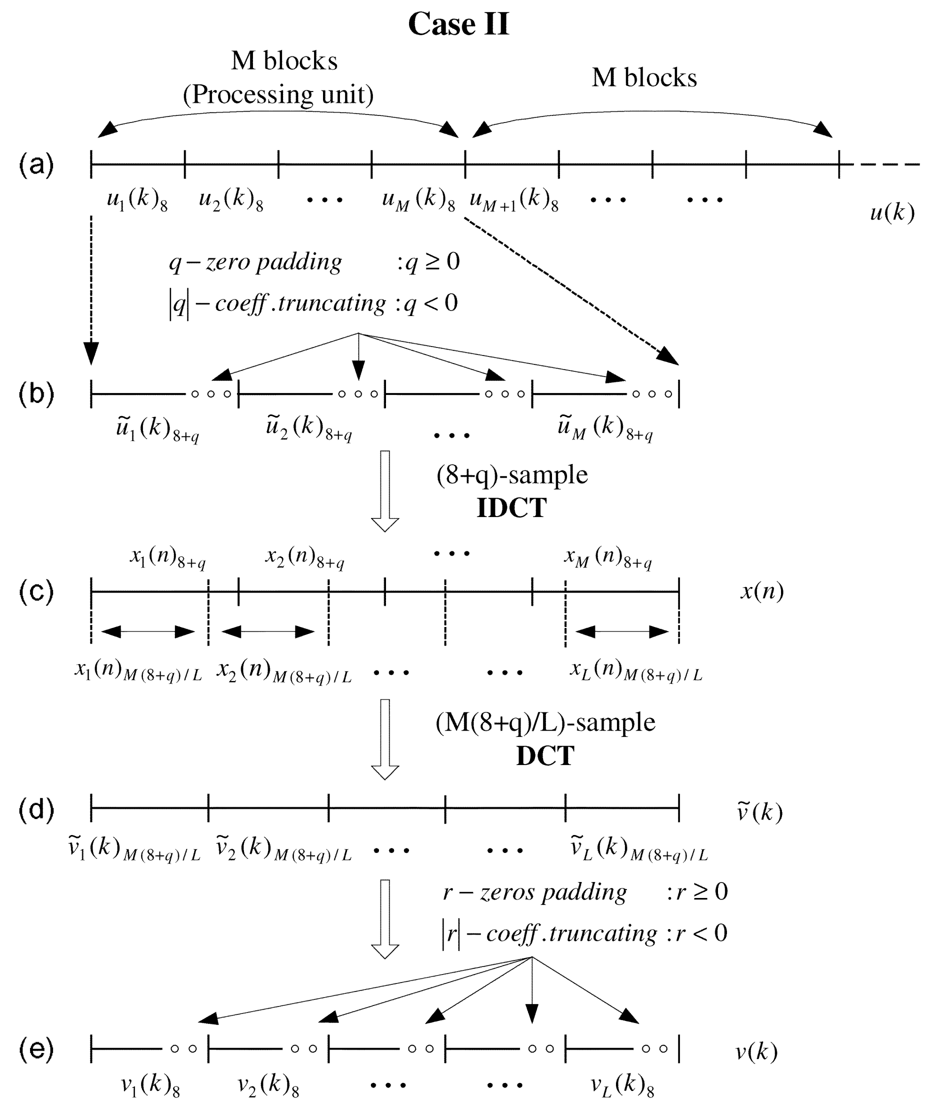


Fig. 2. Conceptual diagram of the faster $L/M$-fold resizing (Case II): (a) processing unit with $M$ blocks, (b) zero padding or truncating in the DCT domain, (c) $(8+q)$-sample IDCT for $M$ blocks and splitting $M$ blocks into $L$ blocks in spatial domain, (d) $(M(8+q)/L)$-sample DCT for$L$ blocks, and (e) resizing by zero padding or truncating high-frequency parts in the DCT domain.

9-sample DCTs for four blocks are performed sequentially. One highest frequence coefficient is truncated for each block after the 9-sample DCT.

If $L/M$ is an integer (integer-ratio upsizing) or $M/L$ is an integer (integer-ratio downsizing), the proposed resizing method of Case I is the same as our previous method [4]. Thus, Case I is the generalized resizing method of [4].

## III. Faster Resizing Method (Case II)

In this section, we describe a faster resizing method based on our proposed method in Section II. The faster method of Case II is also processed with a group of $M$-blocks for the $L/M$-fold resizing. In order to reduce the number of samples to be transformed, Case II adopts the following constraint III instead of constraint II, i.e., constraints I and III are applied to Case II

$$\mathrm{Max}\{M,L\}\times 8 > \tilde{N} \geq \mathrm{Min}\{M,L\}\times 8;\ \mathrm{Constraint\ III}. \quad (7)$$

If there is no $\tilde{N}$ satisfying constraint III, the resizing can be implemented only by using Case I. Since $\tilde{N}$ is determined to be smaller than that of Case I, Case II is computationally faster than Case I. A conceptual diagram of Case II is shown in Fig. 2. $M$ 8-sample blocks are also grouped as shown in Fig. 2(a). In Fig. 2(b), the absolute value of $q$ is the smallest integer value satisfying constraints I and III, and is defined as follows:

$$\tilde{N} = M \times (8+q) \quad (8)$$

where $q$ can be a positive or negative integer value. If $q$ is a negative integer, $q$ high-frequency coefficients are truncated and $(8-|q|)$-sample IDCT is followed in the case of $L/M$-fold downsizing $(M > L)$. If $q$ is a positive integer, $q$ zeros are appended and $(8+q)$-sample IDCT is followed. Since the sample number of the resized block should become eight, zero padding or truncating is also required at the last step. Let $r$ be the required number of padded zeros or the number of truncated coefficients per block at the last step, which can be expressed by

$$\begin{aligned} r &= \{(L\times 8) - \tilde{N}\}/L \\ &= \{(L\times 8) - (M\times(8+q))\}/L. \end{aligned} \quad (9)$$

TABLE I
PSNRs OF VARIOUS RESIZING METHODS FOR THE $3/2$-FOLD UPSIZED IMAGES AFTER $2/3$-FOLD DOWNSIZING

| Method (down:up) (2/3 : 3/2) | Lena | Boat | Peppers | Baboon |
|---|---|---|---|---|
| Bilinear / Bilinear | 33.19 | 30.80 | 32.10 | 24.21 |
| Shu [3] / Case I | 38.81 | 35.46 | 36.15 | 27.04 |
| Previous / Previous [4 ] | 38.73 | 35.40 | 36.12 | 27.01 |
| Case I / Case I | 38.81 | 35.46 | 36.15 | 27.04 |
| Case II / Case II | 38.80 | 35.52 | 36.19 | 27.08 |

TABLE II
PSNRs OF VARIOUS RESIZING METHODS FOR THE $4/3$-FOLD UPSIZED IMAGES AFTER $3/4$-FOLD DOWNSIZING

| Method (down:up) (3/4 : 4/3) | Lena | Boat | Peppers | Baboon |
|---|---|---|---|---|
| Bilinear / Bilinear | 34.22 | 31.74 | 33.14 | 24.78 |
| Previous / Previous [4 ] | 40.45 | 37.47 | 37.61 | 28.54 |
| Case I / Case I | 40.74 | 37.80 | 37.77 | 28.70 |
| Case II / Case II | 40.52 | 37.62 | 37.62 | 28.61 |

If $r$ is a positive integer, $r$ zeros are appended to make eight samples for each block, whereas $|r|$ high-frequency coefficients are truncated if $r$ is a negative integer.

We show an example of $3/4$-fold downsizing using Case II. The total number of samples in a group is 32. Since the number of 32 is not a multiple of $L = 3$, truncating two coefficients per block $(q = -2)$ is required to satisfy constraints I and III. As a result, $\tilde{N}$ is 24. Then, each block is transformed by a 6-sample IDCT and retransformed by a 8-sample DCT.

In the case of $4/3$-fold upsizing, the total sample number $(N)$ of three blocks is 24 as noted above. This also is a common multiple of 4 and 3. This case can be composed of an 8-sample IDCT and a 6-sample DCT. First, the 8-sample IDCT is performed, and the 6-sample DCT is followed sequentially. Two zeros padding per resized block is required at the last step.

## IV. EXPERIMENTAL RESULTS

We compare PSNRs of the proposed method and the previous methods for various images obtained by $2/3$-fold resizing and $3/4$-fold resizing. The PSNRs are measured between the original image and $M/L$-fold upsized image after $L/M$-fold downsizing of the original image. Table I shows the PSNRs of the $3/2$-fold upsized images after $2/3$-fold downsizing by using the previous and the proposed methods. Table II also shows the PSNRs of the $4/3$-fold upsized images after $3/4$-fold downsizing by using the previous and the proposed methods. Our previous paper presented the characteristics of the various resizing filters [8]. The proposed method in the present paper is based on the previous image resizing method [4], [8]. The proposed resizing filter also has a good frequency response. Thus, the PSNR of our method is better than that of the previous methods for various images.

TABLE III
PSNRs OF THE $3/2$-FOLD UPSIZED IMAGES AFTER $2/3$-FOLD DOWNSIZING ACCORDING TO THE VARIOUS $\tilde{N}$

| $\tilde{N}$ | 2/3 : 3/2 (down:up) | | | |
|---|---|---|---|---|
| | Lena | Boat | Peppers | Baboon |
| 12 | 34.66 | 31.50 | 32.73 | 24.19 |
| 18 Case II | 38.80 | 35.52 | 36.19 | 27.08 |
| 24 Case I | 38.81 | 35.46 | 36.15 | 27.04 |
| 30 | 38.79 | 35.45 | 36.15 | 27.04 |
| 36 | 38.76 | 35.43 | 36.14 | 27.02 |
| 42 | 38.74 | 35.41 | 36.13 | 27.01 |
| 48 Previous[4] | 38.73 | 35.40 | 36.12 | 27.01 |
| 54 | 38.71 | 35.39 | 36.12 | 27.00 |
| 60 | 38.71 | 35.38 | 36.11 | 27.00 |

Cases I and II have the same Constraint I. The difference of Case I and Case II is $\tilde{N}$, i.e., the value of $q$ in (2) and (8). We can select various $\tilde{N}$ satisfying (2) or (8). In case of $3/4$-fold downsizing, $\tilde{N}$ of Case I is 36, and Case II is 24 and the previous method [4] is 96. We also present the PSNRs according to $\tilde{N}$ for various images obtained by $2/3$-fold and $3/4$-fold resizing. The PSNRs are measured between the original image and $M/L$-fold upsized image after $L/M$-fold downsizing of the original image. Table III shows the PSNRs of the $3/2$-fold upsized images after $2/3$-fold downsizing according to $\tilde{N}$. Table IV also shows the PSNRs of the $4/3$-fold upsized images after $3/4$-fold downsizing according to $\tilde{N}$. If $\tilde{N}$ is smaller than that of the Case II, PSNR is seriously degraded. As $\tilde{N}$ increases from that of Case I or II, PSNR is slightly decreasing. Table III and Table IV show that the proposed method is the optimum.

We also analyzed the computation amount of multiplications and additions for various resizing methods. Loeffler *et al.* proposed a fast 8-sample DCT algorithm [6], and fast algorithms for various-radix DCTs were developed [7], [9]. The proposed method was implemented with the fast DCT and IDCT introduced in [6], [7], and [9]. In the $3/4$-fold downsizing of Case I, it is the combination of a 9-sample IDCT and a 12-sample DCT. The computation of the 9-sample IDCT requires 8 multiplications and 34 additions, and the computation of the 12-sample DCT requires 14 multiplications and 49 additions [7]. We shall assume that the size of the original image is 16 $8 \times 8$ blocks $(= 32 \times 32)$ and the resized image is nine $8 \times 8$ blocks $(= 24 \times 24)$ for $3/4$-fold downsizing. The amount of computation in 1-D is computed as follows. First, one zero is padded into each 8-sample block, and 9-sample inverse DCT is performed on each zero-padded block. Then, 36 samples of four 9-sample blocks are divided into three blocks with 12 samples. Three

TABLE IV
PSNRs OF THE $4/3$-FOLD UPSIZED IMAGES AFTER $3/4$-FOLD DOWNSIZING ACCORDING TO THE VARIOUS $\tilde{N}$

| $\tilde{N}$ | 3/4 : 4/3 (down:up) | | | |
|---|---|---|---|---|
| | Lena | Boat | Peppers | Baboon |
| 12 | 31.66 | 28.91 | 30.43 | 22.64 |
| 24<br>Case II | 40.52 | 37.62 | 37.62 | 28.61 |
| 36<br>Case I | 40.74 | 37.80 | 37.77 | 28.70 |
| 48 | 40.61 | 37.65 | 37.70 | 28.62 |
| 60 | 40.53 | 37.57 | 37.66 | 28.58 |
| 72 | 40.49 | 37.52 | 37.64 | 28.56 |
| 84 | 40.47 | 37.49 | 37.62 | 28.55 |
| 96<br>Previous[4] | 40.45 | 37.47 | 37.61 | 28.54 |
| 108 | 40.44 | 37.46 | 37.61 | 28.53 |

12-sample DCT is performed on each 12-sample block. Consequently, the computation amount in 1-D is given as follows:

$$\text{Number } of \text{ multiplications} = 74$$
$$= 8 \times 4\ (9 - \text{sample inverse DCT for 4 blocks}) + 14 \times 3\ (12\text{–sample DCT for 4 blocks})$$
$$\text{Number of additions} = 283$$
$$= 34 \times 4\ (9 - \text{sample inverse DCT for 4 blocks}) + 49 \times 3 (12 - \text{sample DCT for 4 blocks}).$$

The 1-D $3/4$-fold downsizing is performed in vertical direction along 32 columns and then in horizontal direction along 24 rows for 2-D image resizing. Therefore, the computation amount for 3/4-fold downsizing of Case I in 2-D is $4.05(= (74 \times 32 \text{ columns} + 74 \times 24 \text{ rows})/(32 \times 32))$ multiplications and $15.48(= (283 \times 32 \text{ columns } + 283 \times 24 \text{ rows})/(32 \times 32))$ additions per pixel of the original image. And the computation amount for $4/3$-fold upsizing of Case I in 2-D is $4.05(= (74 \times 24 \text{ columns} + 74 \times 32 \text{ rows})/(32 \times 32))$ multiplications and $15.48(= (283 \times 24 \text{ columns} + 283 \times 32 \text{ rows})/(32 \times 32))$ additions per pixel of the upsized image.

The computation amounts of $4/3$-fold and $3/2$-fold upsizing for Case II are as follows. The computation of the 8-sample DCT requires 11 multiplications and 29 additions [6], and the computation of the 6-sample IDCT requires four multiplications and 16 additions [9]. The computation amount for $4/3$-fold upsizing of Case II in 2-D is $2.68(= ((33 + 16) \times 24 \text{ columns} + (33 + 16) \times 32 \text{ rows})/(32 \times 32))$ multiplications and $8.26(= ((87+64)\times 24 \text{ columns}+(87+64)\times 32 \text{ rows})/(32\times 32))$ additions per pixel of the original image. $3/2$-fold upsizing is combined of the 9-sample IDCT and 6-sample DCT. The computation amount for $3/2$-fold upsizing of Case II in 2-D is $1.94(= ((16 + 12) \times 16 \text{ columns} + (16 + 12) \times 24 \text{ rows})/(24 \times 24))$ multiplications and $8.06(= ((68 + 48) \times 16 \text{ columns} + (68 + 48) \times 24 \text{ rows})/(24 \times 24))$ additions per pixel of the upsized image.

TABLE V
COMPUTATION AMOUNT OF VARIOUS RESIZING METHODS FOR $L/M$-FOLD DOWNSIZING $(M > L)$

Per pixel of the original image.

| Method | 2/3-fold downsizing | | 3/4-fold downsizing | |
|---|---|---|---|---|
| | Multipli cations | additions | Multipli cations | additions |
| Bilinear | 4.53 | 12.14 | 6.05 | 13.52 |
| Shu | 21.78 | 19.56 | Not available | Not available |
| Previous [4] | 10.42 | 12.50 | 13.02 | 12.91 |
| Case I | 4.24 | 12.85 | 4.05 | 15.48 |
| Case II | 1.94 | 8.06 | 2.68 | 8.26 |

TABLE VI
COMPUTATION AMOUNT OF VARIOUS RESIZING METHODS FOR $L/M$-FOLD UPSIZING $(M < L)$

Per pixel of the original image.

| Method | 3/2-fold upsizing | | 4/3-fold upsizing | |
|---|---|---|---|---|
| | Multipli cations | additions | Multipli cations | Additions |
| Bilinear | 6.19 | 12.14 | 5.61 | 13.52 |
| Previous [4] | 9.79 | 11.94 | 16.63 | 16.19 |
| Case I | 4.24 | 12.85 | 4.05 | 15.48 |
| Case II | 1.94 | 8.06 | 2.68 | 8.26 |

This demonstrates that Case II is computationally faster than Case I. However, the image quality of Case II is slightly lower than or similar to that of Case I. Table V shows the computation amount of the proposed and the previous methods for $2/3$-fold and $3/4$-fold downsizing. In Table V, $3/4$-fold downsizing cannot be implemented by Shu and Chau's method, since $(4/3) \times 8$ is not an integer value. Table VI shows the computation amount of the proposed and the previous methods for $3/2$-fold and $4/3$-fold upsizing.

## V. CONCLUSION

We proposed a fast arbitrary-ratio image-resizing method in the DCT domain. The proposed method is implemented through a combination of inverse and forward DCT of composite lengths. It is easily extended to arbitrary resizing ratio.

The proposed method produces visually fine images and improves the PSNR of the resized images. We also demonstrate that the computational complexity of the proposed method is better than those of the previous methods.

## References


[1] R. Dugad, "A fast scheme for image size change in the compressed domain," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 4, pp. 461–474, Apr. 2001.

[2] J. Mukherjee and S. K. Mitra, "Image resizing in the compressed domain using subband DCT," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 7, pp. 620–627, Jul. 2002.

[3] H. Shu and L. P. Chau, "An efficient arbitrary downsizing algorithm for video transcoding," *IEEE Trans. Circuits Syst. Video Technol*, vol. 14, no. 6, pp. 887–891, Jun. 2004.

[4] H. W. Park, Y. S. Park, and S. K. Oh, "$L/M$-fold image resizing in block-DCT domain using symmetric convolution," in *IEEE Trans. Image Process.*, vol. 12, Sep. 2003, pp. 1016–1034.

[5] C. L. Salazar and T. D. Tran, "On resizing images in the DCT domain," in *Proc. IEEE Int. Conf. Image Processing*, Singapore, Oct. 2004, pp. 2797–2800.

[6] C. Loeffler, A. Ligtenberg, and G. S. Moschytz, "Practical fast 1-D DCT algorithms with 11 multiplications," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. 2, May 1989, pp. 988–991.

[7] G. Bi and L. W. Yu, "DCT algorithms for composite sequence length," *IEEE Trans. Signal Process.*, vol. 46, no. 3, pp. 554–562, Mar. 1998.

[8] Y. S. Park and H. W. Park, "Design and analysis of an image resizing filter in the block-DCT domain," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 2, pp. 274–279, Feb. 2004.

[9] Y. H. Chan and W. C. Siu, "Mixed-radix discrete cosine transform," *IEEE Trans. Signal Process.*, vol. 41, no. 11, pp. 3157–3161, Nov. 1993.



**Young Seo Park** received the B.S. degree in electrical engineering from Yonsei University, Seoul, Korea, in 1990, M.S. degree in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Seoul, in 1998 and the Ph.D. degree from the Department of Electrical Engineering, KAIST, Daejeon, Korea, in 2005.

He was a Researcher at Samsung Electronics Company, Ltd., Suwon, Korea, from 1990 to 1999. He is now with Ontimetek, Sungnam-si, Kyunggi-do, Korea. His current research interests include video coding systems, compressed domain image manipulation, and image computing systems.



**Hyun Wook Park** (SM'99) received the B.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 1981 and the M.S. and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Seoul, in 1983 and 1988, respectively.

He was a Research Associate at the University of Washington from 1989 to 1992 and a Senior Executive Researcher at Samsung Electronics Company, Ltd., Suwon, Korea, from 1992 to 1993. Since 1993, he has been a Professor in the Electrical Engineering Department of KAIST. His current research interests include image computing systems, image compression, medical imaging, and multimedia systems.

Dr. Park served as an Associate Editor for the *International Journal of Imaging Systems and Technology*. He is a member of SPIE.