

# DCT 영역에서 영상의 임의 비율 크기 변환을 위한 새로운 알고리즘

준회원 김 용 재\*, 정회원 이 창 우\*\*

## New Algorithm for Arbitrary-ratio Image Resizing in DCT Domain

Yong-Jae Kim\* Associate Member, Chang-Woo Lee\*\* Regular Member

### 요 약

유비쿼터스 통신 환경에서는 영상의 다양한 형태의 변환이 필수적인데, 대부분의 디지털 영상은 DCT (Discrete Cosine Transform)를 기반으로 한 JPEG, MPEG 등의 표준 기법을 이용하여 압축되어 저장되어 있다. 본 논문에서는 DCT 영역에서 영상의 크기를 변환시키는 기존의 여러 가지 알고리즘들을 고찰하고 그 성능을 분석하며, 기존의 방법보다 더 우수한 성능을 보이는 DCT 영역에서의 영상 크기 변환 알고리즘을 제안한다. 제안하는 알고리즘은 DCT 영역에서 영상의 임의 크기 변환을 위해 여러 개의  $8 \times 8$  DCT 계수 블록을 변환 비율에 따라 변환식을 통하여 하나의 블록으로 변환하고 최적의 zero-padding 및 truncating을 위한 IDCT의 크기를 정하는 방법을 이용하여 영상 크기 변환의 성능을 높인다. 이것은 화소간의 상관도를 최대한 이용하여 DCT 계수를 구하고 여기서 얻어진 DCT 계수 블록에서 원하는 비율에 따라 최적의 크기를 구함으로써 성능을 높이는 알고리즘이다. 그 성능을 원 영상과 축소하여 다시 확대시킨 영상의 PSNR 비교를 통하여 확인하였고, 제안하는 알고리즘은 특정 비율의 변환에 있어서 기존의 알고리즘을 포괄할 수 있는 방법임을 확인할 수 있었다.

**Key Words** : DCT Domain, Arbitrary-ratio Image Resizing, Subband DCT, Composite Length DCT

### ABSTRACT

In Ubiquitous communication environment, various conversions of images are essential, and most digital images are compressed by standard methods such as the Joint Photographic Expert Group (JPEG) and Motion Picture Expert Group (MPEG) which are based on the discrete cosine transform (DCT). In this paper, various image resizing algorithms in the DCT domain are analyzed, and a new image resizing algorithm, which shows superior performance compared with the conventional methods, is proposed. For arbitrary-ratio image resizing in the DCT domain, several blocks of  $8 \times 8$  DCT coefficients are converted into one block using the conversion formula in the proposed algorithm, and the size of the inverse discrete cosine transform (IDCT) is decided optimally. The performance is analyzed by comparing the peak signal to noise ratio (PSNR) between original images and converted images. The performance of the proposed algorithm is better than that of the conventional algorithm, since the correlation of pixels in images is utilized more efficiently.

### I. 서 론

유비쿼터스 통신 환경에서는 사용자가 처한 환경

에 따라서 영상을 포함한 멀티미디어 콘텐츠를 다양한 형태로 변환하는 것이 필수적이다. 대부분의

※본 연구는 2006년도 가톨릭 대학교 교비 연구비 지원으로 이루어졌습니다.

\* 가톨릭대학교 대학원 컴퓨터공학과 신호처리 연구실, \*\* 가톨릭대학교 정보통신전자공학부 (changwoo@catholic.ac.kr)

논문번호 : KICS2006-08-340, 접수일자 : 2006년 8월 10일, 최종논문접수일자 : 2007년 1월 31일

디지털 영상들은 JPEG, MPEG과 같은 표준 기법을 사용하여 압축되어 저장되는데 이는 DCT (Discrete Cosine Transform)를 기반으로 하고 있다. 따라서 DCT 영역에서 영상의 크기를 직접 변환할 수 있는 방법이 그 효율성으로 인해 많이 연구되고 있다<sup>1-3, 5-7</sup>.

영상의 크기를 DCT 영역에서 직접 변환하기 위해서 Dugad 등은 8×8 DCT 블록의 고주파 계수들을 제거함으로써 영상의 크기를 축소하는 방법 (truncating)과, 고주파 부분에 0을 삽입함으로써 영상의 크기를 확대하는 방법 (zero-adding)을 제안하였다<sup>11</sup>. 이것을 기반으로 subband DCT를 이용하는 방법과 블록과 블록 간의 DCT 계수 값의 관계를 이용하는 방법이 제안되었고, 최근에는 DCT 영역에서 임의 비율의 영상 크기 변환을 위한 알고리즘이 많이 연구되고 있다<sup>12-17</sup>. 그리고 이러한 최근의 알고리즘들은 영상의 상관도를 최대한 이용하기 위해 보다 많은 화소를 이용하고, IDCT 과정에서 최적의 크기를 이용함으로써 초기 알고리즘들에 비해 향상된 성능을 보인다.

본 논문에서는 DCT 영역에서 영상을 임의 크기로 변환할 때, 포괄적이면서 보다 개선된 성능을 얻을 수 있는 알고리즘을 제안한다. 제안하는 알고리즘의 원리는 DCT 영역에서 영상의 임의 크기 변환을 위해 원 영상의 여러 개의 8×8 DCT 계수 블록을 변환비율에 따라 모아 변환식을 통하여 변환하고 최적의 zero-padding 및 truncating을 위한 IDCT 크기를 정하는 방법으로 영상의 성능을 높인다. 먼저, L/M 비율의 축소 변환의 경우를 보면 원 영상의 M×M 개의 8×8 블록의 DCT 계수를 변환식을 이용하여 (8×M)×(8×M) 크기의 DCT 계수를 가지는 블록으로 변환시킨다. 이렇게 얻어진 DCT 계수 블록을 IDCT의 크기에 맞추어 0의 삽입(zero-padding) 또는 고주파 계수를 제거(truncating)하고 (8×L)-point IDCT와 8-point DCT를 함으로써 우리가 원하는 변환된 영상 크기의 8-point DCT 계수를 얻을 수 있다. 마찬가지로, M/L 비율의 확대 변환의 경우에는 먼저 원 영상의 L×L개의 8×8 DCT 계수 블록을 변환식을 이용하여 원 영상의 (8×L)-point DCT의 효과를 얻을 수 있는 DCT 계수를 구한다. L/M 비율의 축소 변환에서와 같은 방법으로 IDCT 크기에 맞추어 0의 삽입 (zero-padding) 또는 고주파 계수를 제거(truncating)하고 (8×M)-point IDCT와 8-point DCT의 과정을 거쳐 확대 변환된 영상의 크기를 얻을 수 있다. 즉, 제안하는 알고리즘은

Mukherjee 등이 제안한 변환행렬 과 Park 등이 제안한 알고리즘에서 사용한 composite length DCT를 이용하여<sup>17-18</sup>, DCT 영역에서 영상의 크기를 변환함으로써 특정 비율에 있어 기존의 방법을 포괄할 수 있는 알고리즘이다. 이와 같이, 제안하는 알고리즘은 영상의 화소간의 상관도를 최대한 이용하기 위해 변환식을 사용하고, 최적의 IDCT 크기를 구현함으로써 기존의 알고리즘보다 성능을 향상시키는 알고리즘이다.

본 논문의 구성은 다음과 같다. 본 논문의 2장에서는 지금까지 알려져 있는 DCT 영역에서 영상의 크기 변환을 위한 기존의 알고리즘들을 설명하고, 3장에서는 DCT 영역에서 영상의 임의 크기 변환을 위한 새로운 알고리즘을 제안한다. 그리고 4장에서 영상의 크기 변환을 위한 다양한 알고리즘들의 성능측정을 위하여 PSNR 결과를 제시하고 계산량 비교를 통하여 복잡도를 비교한다. 마지막으로 5장에서 결론을 맺는다.

## II. DCT 영역에서 영상 크기 변환을 위한 기존의 알고리즘

Dugad 등이 제안한 영상의 크기를 DCT 영역에서 직접 변환하는 알고리즘을 설명하면 다음과 같다<sup>11</sup>. 1/2배 축소 변환의 경우 그림 1에서 도시하는 것처럼, 먼저 원 영상의 8×8 DCT 블록 4개를 각각 B<sub>11</sub>, B<sub>12</sub>, B<sub>21</sub>, B<sub>22</sub>라 하고, 이것을 하나로 묶은 {B<sub>11</sub>, B<sub>12</sub>, B<sub>21</sub>, B<sub>22</sub>}를 B라고 하자. 각각의 블록에서 4×4 크기의 저주파 DCT 계수 값만을 남기고 나머지 고주파 DCT 계수 값은 제거한 후 4-point IDCT한 결과를  $\hat{b}_{11}$ ,  $\hat{b}_{12}$ ,  $\hat{b}_{21}$ ,  $\hat{b}_{22}$ 라 하고, { $\hat{b}_{11}$ ,  $\hat{b}_{12}$ ,  $\hat{b}_{21}$ ,  $\hat{b}_{22}$ }와 같이 묶은 8×8 블록을  $\hat{b}$ 라 하자. 이렇게 얻은  $\hat{b}$ 를 8-point DCT로 연산한 결과인  $\hat{B}$ 가 가로 세로 방향으로 1/2 비율로 축소된 결과이다. 또한, 확대 변환의 과정은 다음과 같다. 축소된 영상의 공간영역에서의 8×8 블록을 4×4 블록 4개로 나눈 각각에 대해서 4-point DCT를 취한 것을  $\hat{B}_{11}$ ,  $\hat{B}_{12}$ ,  $\hat{B}_{21}$ ,  $\hat{B}_{22}$ 라 하자. 여기서 얻어진 각각의

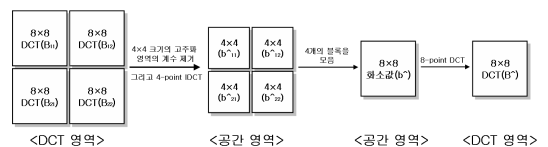


그림 1. Dugad 등에 의해 제안된 1/2 비율의 영상 크기 변환 Fig. 1. 1/2 fold image resizing proposed by Dugad

블록에 대해 8×8 크기에 해당되도록 나머지 고주파 부분에 0을 삽입한 후 8-point IDCT를 하면 확대 변환된 영상을 구할 수 있다. 즉, 확대된 영상의 한 블록인  $\hat{C}_{11}$ 을  $\{\hat{B}_{11}, 0, 0, 0\}$ 라 할 때 0은 4×4 크기의 0행렬이고, 나머지  $\hat{C}_{12}, \hat{C}_{21}, \hat{C}_{22}$ 도 같은 방법으로 얻을 수 있다. 이러한 영상의 크기 변환을 위해서 고주파 성분을 제거하거나 고주파 성분에 0을 삽입하는 것은 DCT 영역에서 영상의 크기를 임의의 비율로 변환시키는 다른 알고리즘들의 기본 원리가 된다.

DCT 영역에서 영상의 크기를 임의의 비율로 변환하기 위해서 Salazar, Mukherjee 등은 그림 2와 3에서 각각 도시한 것과 같은 방법을 제안하였다<sup>[5-6]</sup>. 먼저 Salazar 등에 의해서 제안된 임의의 비율로 영상의 크기를 변환시키는 알고리즘을 그림 2에 도시하였다<sup>[6]</sup>.

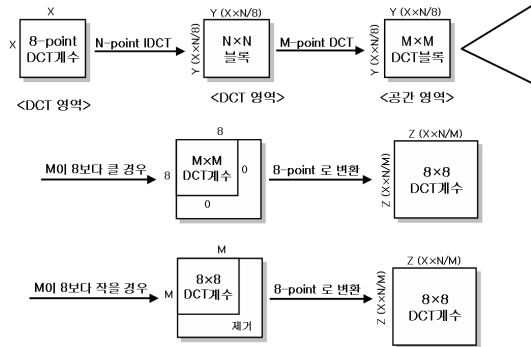


그림 2. Salazar 등에 의해 제안된 임의의 N/M 비율의 영상 크기 변환  
Fig. 2. Image resizing of Arbitrary N/M ratio proposed by Salazar

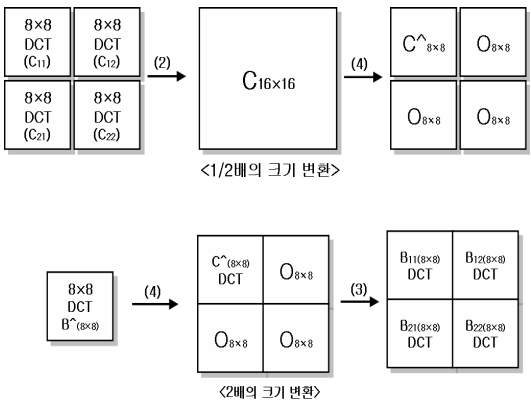


그림 3. 1/2배와 2배에 대해 Mukherjee 등이 제안한 알고리즘  
Fig. 3. The Algorithm proposed by Mukherjee for image halving and doubling

원 영상의 크기를 X×X라 할 때, 이것을 DCT 영역의 8×8 블록으로 나누어진 것을 각 8×8 블록마다 N-point IDCT를 수행하여 영상의 크기를 조절한다. 여기서 N<8일 경우 N개만큼의 DCT 계수 값을 남기고 나머지는 제거한다. 그리고 N>8일 경우 8×8 DCT 계수 값은 그대로 유지하고 (N-8)개만큼의 0을 삽입한다. 이 과정에서 X×X 크기의 원 영상의 크기는 Y×Y((X×N/8)×(X×N/8)) 크기로 변환되게 된다. 다음의 과정인 M-point DCT와 고주파 성분의 제거 혹은 고주파 성분에 0을 삽입함으로써 M-point 수를 8-point의 수로 맞추는 과정을 통해 구하고자 하는 영상의 크기 Z×Z((X×N/M)×(X×N/M))의 DCT 계수 블록을 얻을 수 있다. 그 다음, Mukherjee 등이 제안하는 알고리즘은 Jiang & Feng이 제안한 블록과 그 하위 블록 사이의 DCT 계수 값 간의 관계와 subband DCT를 이용한 방법인데<sup>[3-5]</sup>, subband DCT에 대해 알아보면 먼저, 1/2 비율에 대한 subband DCT는 다음 식 (1)의 관계를 이용하여 구현된다<sup>[3]</sup>.

$$C_{N \times N}(k, l) = \begin{cases} 4C_{N/2 \times N/2}(k, l), & (k, l) = 0, \dots, N/2 - 1 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Dugad 등에 의해 제안된 알고리즘에서는 고주파 부분의 계수를 제거한 후 4-point IDCT와 8-point DCT를 통하여 축소된 영상을 얻지만, subband DCT를 이용한 알고리즘에서는 (1)의 식을 이용하여 8×8 크기의 DCT 블록을 4×4 크기의 DCT 블록으로 변환한 후, 4-point IDCT와 8-point DCT를 이용하여 크기를 변환하는 방법상의 차이를 보인다. 또한, 2배의 비율로 확대 변환하는 방법은 다음과 같은 차이를 보인다. Dugad 등이 제안한 2배의 확대 변환의 경우 원 영상의 8×8 DCT 계수를 4×4 DCT 계수로 변환한 후, 4×4 DCT 계수 한 블록에 고주파 성분을 0의 값으로 삽입하여 8×8 크기의 DCT 계수로 변환한다. 그러나 subband DCT를 이용한 알고리즘에서는 8×8 DCT 계수 한 블록에 식 (1)을 통하여 16×16 크기의 DCT 계수 블록으로 변환하고 변환된 블록을 16-point IDCT와 8-point DCT를 통하여 확대된 영상의 8×8 DCT 계수를 얻는다는 점에서 차이를 보인다. 원 영상들을 1/2 비율로 크기를 축소 변환시킨 후 다시 2배의 비율로 영상의 크기를 확대 변환하여 원 영상과 비교한 PSNR 값에서, 확대과정에서 더 많은 공간 상관도를 이용하여 구현한 subband DCT의 알고리즘

표 1. 1/2배로 축소변환 후 2배로 확대 변환하여 원 영상과 비교한 PSNR 값  
Table 1. PSNR values after image halving and doubling

Algorithms	PSNR[dB]		
	lena	f-16	boats
Mukherjee[3]	34.80	34.02	32.61
Dugad[1]	34.49	33.64	32.52

즘이 더 우수한 성능을 보여 주고 그 결과를 표 1에 제시하여 두 알고리즘의 성능을 비교하였다. Mukherjee 등에 의해 제안된 알고리즘은 위의 subband DCT와 Jiang & Feng이 제안한 DCT 계수와 그 하위 블록의 DCT 계수 값 간의 관계<sup>[4]</sup>를 이용하여 DCT 영역에서 임의 비율로 영상의 크기를 변환시키는 알고리즘으로서 그 과정을 다음의 예로 설명할 수 있다<sup>[5]</sup>. 먼저 1/2 비율로 축소 변환하는 과정에서는 그림 3에 도시된 것처럼 원 영상의 8×8 DCT 블록 4개와 변환행렬 A<sub>(2,8)</sub>를 이용한 식 (2)의 관계를 이용하여 16×16 DCT 블록으로 변환시키고 여기서 고주파 부분의 계수를 제거함으로써 1/2배 축소 변환한다.

$$C_{16 \times 16} = A_{(2,8)} \begin{pmatrix} C_{11(8 \times 8)} & C_{12(8 \times 8)} \\ C_{21(8 \times 8)} & C_{22(8 \times 8)} \end{pmatrix} A^{T(2,8)} \quad (2)$$

일반적으로 A<sub>(M,8)</sub>는 M×M개의 8×8 DCT 계수 블록을 (8×M)×(8×M) 크기의 DCT 계수 블록으로 변환하는 (8×M)×(8×M) 크기의 변환행렬을 의미한다. 따라서 식 (2)는 8×8 DCT 계수 블록 4개를 모아 변환행렬 A<sub>(2,8)</sub>에 의해 하나의 16×16 DCT 계수로 변환시킬 수 있는 식이다. 반면에 2배의 확대 변환은 영상의 8×8 DCT 계수 한 블록을 subband DCT식 (1)에 의해 16×16 DCT 계수 블록으로 변환하고 변환행렬 A<sub>(2,8)</sub>의 역행렬인 A<sup>-1</sup><sub>(2,8)</sub>을 이용하여 8×8 DCT 계수 블록 4개로 변환함으로써 확대할 수 있고 식 (3)에 의해 구현될 수 있다.

$$A_{(2,8)}^{-1} C_{16 \times 16} A_{(2,8)}^{-1 T} = \begin{pmatrix} C_{11(8 \times 8)} & C_{12(8 \times 8)} \\ C_{21(8 \times 8)} & C_{22(8 \times 8)} \end{pmatrix} \quad (3)$$

Mukherjee 등이 제안하는 영상의 임의 비율의 변환방법은 위의 1/2배와 2배의 방법을 기반으로 하고 있다. 이 알고리즘에서는 식 (4)와 같은 subband식을 제안하고 있다.

$$C_{L \times L}(k, l) = \begin{cases} \sqrt{L \times L} \cdot C_{N \times N}(k, l), & (k, l) = 0, \dots, N-1 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

식 (4)에서 L값은 임의 비율에 대한 값이다. 예를 들어 1/3배의 축소 변환을 하려한다면 L의 값은 3이다. 그러나 3/4배의 변환은 L은 3으로 하여 원 영상을 3배의 비율로 확대 변환한 후, 다시 L을 4로 하여 1/4배의 비율로 축소 변환하여 구한다. 그런데 3/4배 혹은 2/3배와 같은 비율로 영상의 크기 변환을 하기 위해서는 확대의 과정을 한 후 다시 축소의 과정을 거쳐야 하기 때문에 계산량이 증가하며 알고리즘을 두 번 시행해야 원하는 비율의 영상 크기를 구현할 수 있다는 단점이 있다. 예를 들어 3/4 비율에 대한 과정을 보면, 먼저 3배의 확대 변환을 위해 원 영상의 8×8 DCT 계수 한 블록을 subband DCT식 (4)에 의해 24×24 DCT 계수 블록으로 변환하고 변환행렬 A<sub>(3,8)</sub>의 역행렬인 A<sup>-1</sup><sub>(3,8)</sub>을 이용하여 8×8 DCT 계수 블록 9개로 변환함으로써 확대할 수 있다. 이와 같은 과정에 의해 확대 변환된 영상을 다시 1/4배의 비율로 축소 변환하기 위해 확대 변환된 영상의 8×8 DCT 계수 블록 16개를 변환행렬 A<sub>(4,8)</sub>를 이용하여 32×32 DCT 계수 블록으로 변환시키고 여기서 고주파 부분의 계수를 제거하여 1/4배 축소 변환함으로써, 두 번의 알고리즘 실행을 통해 구현하고자 하는 3/4 비율의 변환된 영상의 크기를 얻을 수 있다.

가장 최근에 제안된 Park 등이 제안한 알고리즘은 N/M 비율에 대해 Salazar 등이 제안한 알고리즘을 기반으로 하고, composite length DCT를 이용하여 최적의 DCT와 IDCT의 크기를 정함으로써 임의 비율 변환을 보다 효율적으로 구현한 것이다<sup>[7-8]</sup>. 예를 들어 Salazar 등이 제안한 알고리즘을 이용하여 2/3 비율로 영상의 크기를 변환하고자 한다면 N을 2, M을 3으로 하거나 N을 4, M을 6으로 하는 등 여러 가지 값을 할당하여 구할 수 있지만, Park 등이 제안하는 알고리즘에서는 2/3 비율의 영상 크기 변환을 위한 최적의 N과 M의 값을 다음과 같은 방법을 이용하여 구한다. L/M(L과 M은 더 이상 약분될 수 없는 수로 1/2의 비율일 경우 L은 1, M은 2와 같다.)의 임의 비율에 대해서 N'은 M과 L의 공배수라는 조건과 N'이 M과 L중 큰 값에 8을 곱한 값보다 크거나 같은 값이라는 조건에 맞추어 N'를 구하고 N'=M×(8+q)의 식에 의해 q를 구한다. 단, q는 0을 포함하여 가장 작은 음이 아닌 정수이다(q ≥ 0). 여기서 q를 얻게 되면 Salazar 등이 제안한 알고리즘에서 N-point IDCT의 N값 대신에 (8+q)값을 사용하게 된다. 그 다음, [M×(8+q)/L]-point

DCT를 하고 8×8 크기로 맞추면 우리가 구하고자 하는 L/M 비율의 영상을 얻을 수 있다. 예를 들어 3/4배의 축소 변환의 경우에는 L/M에서 L은 3, M은 4라고 할 수 있고 위의 조건에 맞추기 위해 두 번째 조건에서 N은 32의 값을 갖지만, 이 값은 첫 번째 조건에 위배되어 조건에 맞는 가장 가까운 수 36이 선택된다. 따라서 q값은 1이 된다. 그러므로 (8+1)-point IDCT를 하고 이렇게 얻어진 영상에서 M×(8+q)/L을 이용하여 4×(8+1)/3은 12로 12-point DCT를 한다. 그 다음 8×8 크기로 바꾸기 위해 고주파 부분의 계수를 제거하는 과정을 거쳐서 우리가 얻고자하는 3/4배의 영상을 얻을 수 있다. 4/3배의 영상 크기 변환 또한 마찬가지로 구현할 수 있다. 전체적인 알고리즘은 기본적인 원리에 있어서 Salazar 등에 의해 제안된 접근법과 비슷하지만 다음과 같은 차이가 있다. 1/2 비율로 영상의 크기를 변환할 때, Salazar 등에 의해 제안된 방법에서는 3/6, 4/8, 5/10, 6/12 등의 여러 가지 비율의 값을 이용하지만<sup>6</sup>, Park 등에 의해 제안된 방법에서는 좋은 PSNR 값이나, 빠른 계산을 위해 최적의 DCT와 IDCT의 크기를 정함으로써 최적의 비율을 찾는다<sup>7</sup>. 따라서 3/4 비율에서는 16-point IDCT와 12-point DCT를 행하였을 때 3/4 비율의 영상 크기 변환에 대한 최적의 성능을 구현할 수 있음을 보인 것이다.

### III. 제안하는 알고리즘

본 장에서는 DCT 영역에서 임의의 비율로 영상의 크기를 변환하기 위해 기존의 알고리즘보다 더 우수한 성능을 보이며, 특정 비율에 대해 기존 알고리즘을 포괄할 수 있는 새로운 알고리즘을 제안한다. 제안하는 알고리즘은 영상의 상관도를 최대한 이용하기 위해서 보다 많은 화소를 이용하여 DCT 계수를 구하는 것과 이 DCT 계수에서 원하는 비율에 따라 최적의 크기로 영상의 크기 변환을 구현하는 것을 기본 원리로 한다. 즉, 변환 행렬을 이용하여 더 큰 블록의 DCT 계수를 얻을 수 있고, 이 DCT 계수 블록에서 원하는 비율에 맞추어 IDCT의 크기를 정한다. L/M(L<M) 비율의 축소 변환에 대해, 그림 4에서 도시한 것과 같이 제안하는 알고리즘은 먼저, 원 영상의 8-point DCT 계수 블록을 M×M개 모은다. 여기서 얻은 (8×M)×(8×M) 크기의 블록을 변환 행렬 A<sub>(M,8)</sub>에 의해 원 영상의 (8×M)-point DCT를 행한 것과 같은 (8×M)×(8×M) DCT 계수 블

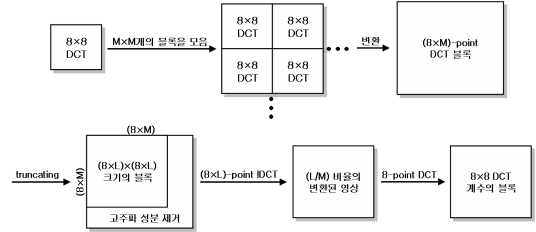


그림 4. L/M 비율의 축소 변환에 대해 제안하는 알고리즘  
Fig. 4. The proposed algorithm for L/M-fold downsizing

록을 얻는다. 이렇게 얻은 DCT 계수 블록들은 아직은 원 영상의 크기를 유지하고 있으므로 이 DCT 계수 블록들에 (8×L)-point IDCT와 8-point DCT를 하여 우리가 원하는 변환된 영상 크기의 8×8 DCT 계수를 구할 수 있다. (8×L)-point IDCT의 과정에서 8×M의 수보다 8×L의 수가 더 작기 때문에 (8×M)×(8×M) DCT 계수 블록의 고주파 성분을 제거한다. 이러한 과정을 거친 것은 더 큰 DCT 계수 블록을 구하여 공간 상관도를 최대한 이용하기 위한 것과 비율에 맞추어 크기 변환을 구현하기 위해 IDCT 크기를 정함으로써 zero-padding 혹은 truncating의 과정을 수행하기 위함이다.

여기서 IDCT의 크기가 (8×L)인 것은 다음의 과정을 통해서 얻어낼 수 있다. 간편하게 1차원의 경우로 살펴보면, 원 정보의 크기 X를 (L/M)×X 크기로 변환해야 하고 이것을 Y라고 하자. 변환행렬 A<sub>(M,8)</sub>에 의해 얻은 (8×M)-point DCT 계수 블록은 원 정보를 (8×M)-point DCT한 것과 같은 효과를 얻는다고 하였으므로 블록 수는 [X÷(8×M)]임을 확인할 수 있고, 아직은 원 정보의 크기를 유지하고 있다. IDCT 과정에서 IDCT의 크기를 N이라고 가정하면, 각 (8×M)×(8×M) DCT 계수 블록에 N-point IDCT를 하게 됨으로 블록의 크기는 N으로 변하게 된다. 그러나 전체 블록 수는 [X÷(8×M)]이므로 영상의 크기는 [X÷(8×M)]×N이다. 이 값이 변환시키고자 하는 크기 Y와 같으므로 식 (5)와 같이 표현할 수 있다.

$$(L/M) \times X = [X / (8 \times M)] \times N \quad (5)$$

식 (5)에서 우변의 N을 남기고 정리하면 N의 값은 다음과 같이 구할 수 있다.

$$N = 8 \times L \quad (6)$$

위의 과정을 통하여 IDCT의 크기 N을 (8×L)로

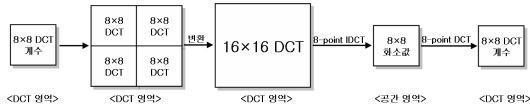


그림 5. 1/2배 축소 변환의 제안하는 알고리즘  
Fig. 5. The proposed algorithm of image halving

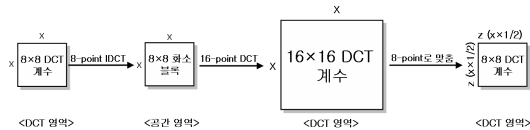


그림 6. 8/16배 축소 변환에 대해 Salazar등이 제안한 알고리즘  
Fig. 6. The algorithm proposed by Salazar for 8/16-fold downsizing

결정할 수 있다. 여기서 임의의 비율을  $L/M$ 으로 표현하고  $L$ 과  $M$ 은 더 이상 약분될 수 없는 수라고 가정한다면, 축소 변환 알고리즘을 정리하면 다음과 같다.

$L/M(L < M)$  비율의 크기 변환을 위한 제안하는 알고리즘(축소 변환의 경우):

- ① 원 영상의  $8 \times 8$  DCT블록을  $M \times M$ 개 모은다.
- ②  $M \times M$ 개의 블록을 모은 것을 변환행렬  $A_{(M,8)}$ 을 이용하여  $(8 \times M)$ -point DCT 블록으로 변환한다.
- ③  $(8 \times M)$ 의 길이와  $(8 \times L)$ 의 길이를 비교하면  $(8 \times M)$ 의 길이가 더 크다. 따라서  $(8 \times M) - (8 \times L)$ 의 차이만큼 고주파 성분을 제거한다.
- ④  $(8 \times L) \times (8 \times L)$  크기의 블록을  $(8 \times L)$ -point IDCT를 하고 DCT 계수로 변환시키기 위해 8-point DCT를 한다.

제안하는 알고리즘에서  $L$ 의 값이 1인 경우를 주목할 필요가 있는데, 제안하는 알고리즘을 이용한 1/2배의 축소 변환을 그 예로 살펴본다. 1/2배 축소 변환의 경우 제안하는 알고리즘에서  $M$ 은 2이므로 그림 5에 도시한 것과 같이 먼저 원 영상의 8-point DCT 계수의 블록을 2x2개 모은다. 모아진 블록을 변환행렬  $A_{(2,8)}$ 을 이용하여 원 영상에서 16-point DCT를 한 것과 같은 DCT 계수 블록으로 변환시킨다. 그 다음 제안한 알고리즘의 과정 ③과 같이 하였을 때,  $(8 \times 2)$ 와  $(8 \times 1)$ 의 차이인 8을 얻게 된다. 이상에서, 전체적인 알고리즘의 도식이 그림 3의 Mukherjee 등에 의해 제안된 1/2 비율의 알고리즘과 유사함을 확인 할 수 있다. 다만, Mukherjee 등이 제안한 알고리즘은 그림 3에 도시한 것과 같이

축소 변환된 영상의 8-point DCT 계수를 얻기 위해 subband DCT식 (4)를 이용하여 구현한 반면에, 제안하는 알고리즘에서는 그림 5에서처럼 8-point IDCT와 DCT의 과정을 거치는 차이를 보인다. 1/2 배 축소 변환뿐만 아니라 1/3 혹은 1/4 등의 축소 변환의 경우에도 제안하는 알고리즘에서  $L$ 의 값이 1로 나타나 같은 비율에 대해서 Mukherjee 등에 의해 제안된 알고리즘과 유사한 모습을 보인다. 이상의 사실에서, Mukherjee 등이 제안하는 알고리즘은 본 논문에서 제안하는 알고리즘의  $L$ 이 1인 특별한 경우로 생각할 수 있다.  $L$ 이 1이 아닐 경우에 대한 Mukherjee 등이 제안하는 구현방법은 2장에서 3/4 비율로 예를 보였다. 즉, 같은 비율에 대해 본 논문에서 제안하는 알고리즘과는 다르게 3배의 확대 변환 후 다시 1/4배의 축소 변환을 통해 구현됨으로  $L$ 의 값이 1인 경우와 다르게 구현될 수 있다.  $L$ 의 값이 1인 경우에 대해 4장에서 제시하는 1/2배와 1/3배의 축소 변환 후 2배와 3배로 각각 확대 변환하여 얻어낸 영상의 성능을 나타내는 PSNR 값을 비교해보면 이 비율에 대한 Mukherjee 등에 의해 제안된 방법과 본 논문에서 제안하는 방법의 결과 값이 유사함을 확인 할 수 있다.

또한, Salazar 등이 제안하는 알고리즘과 본 논문에서 제안하는 알고리즘을 비교하면 다음과 같다. 먼저 Salazar 등이 제안한  $N$ 이 8이고  $M$ 이 16일 때 영상을 축소하는 방법은 그림 6에 도시하였다. 그림 6에서 원 영상의  $8 \times 8$  DCT 블록을 8-point IDCT와  $16 \times 16$  DCT를 하는데, 이렇게 얻은 그림상의 세 번째 블록은 원 영상의 16-point DCT를 하여 얻은 블록과 같다. 이것은 그림 5에서 변환행렬을 이용하여 얻은 세 번째 블록과 같고, 그림 6에서 크기를 8로 맞추는 과정을 제안하는 알고리즘에서는 8-point IDCT와 DCT를 이용한 차이를 보이지만,  $16 \times 16$  DCT에서  $8 \times 8$  DCT 계수로 변환하는 전체적인 흐름에 있어 유사함을 보인다. 그리고 4장에서 제시하는 PSNR 값을 보면 1/2 비율에 대한 본 논문에서 제안하는 방법의 결과 값과  $N$ 을 8,  $M$ 을 16로 할당한 Salazar 등에 의해 제안된 방법으로 실험한 결과 값이 같음을 확인 할 수 있다.

그러나  $L$ 을 1로 고정하고  $M$ 의 값을 3으로 변환하여 구현하였을 때는 완전한 차이를 보인다. 즉, Salazar 등이 제안하는 방법에서는 원 영상의  $8 \times 8$  DCT 계수 블록에  $1 \times 1$  크기로 zero-padding하여 9-point IDCT를 하고 다시 27-point DCT한 후 크

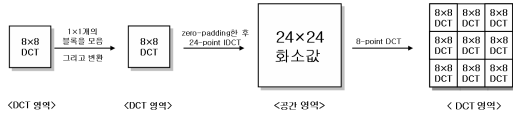


그림 7. 3배 확대 변환에 대해 제안하는 알고리즘  
Fig. 7. The proposed algorithm for 3-fold upsizing

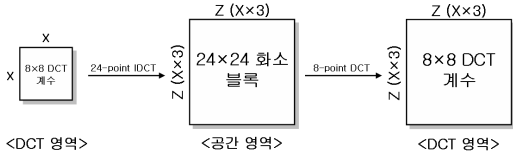


그림 8. 24/8배 확대 변환에 대해 Salazar 등이 제안 한 알고리즘  
Fig. 8. The algorithm proposed by Salazar for 24/8-fold upsizing

기를 8로 맞추지만, 제안하는 알고리즘에서는 원 영상의 8×8 DCT 블록을 3×3개 모아 변환행렬에 의해 24×24 DCT 계수 블록으로 변환하여 8-point IDCT와 DCT를 거치는 상반됨을 보인다. 하지만, 2배와 3배 등의 비율로 확대 변환할 경우(즉, M의 값을 1로 고정할 경우) 제안하는 알고리즘의 포괄성을 확인할 수 있다. 그 예로 그림 7과 8에 도시한 것과 같이 3배의 변환에 대한 제안하는 알고리즘과 Salazar 등이 제안한 N이 24, M이 8일 때에 대한 방법을 비교하였다. 먼저, 그림 7에 도시한 것과 같이 제안하는 알고리즘에서 원 영상의 8×8 DCT 계수 블록에 1×1 크기로 블록을 모아 변환하므로 똑같은 8×8 DCT 계수 블록을 얻게 된다. 따라서 그림 상의 두 번째 블록에서 마지막 블록의 과정까지의 모든 과정이 그림 8에 도시한 Salazar 등이 제안한 알고리즘과 같음을 확인할 수 있다. 즉 8×8 DCT 계수 블록에 16×16 크기로 zero-padding하여 24×24 크기로 변환하고 이 변환된 블록에 24-point IDCT와 8-point DCT의 과정을 거쳐 3배로 확대된 영상의 8×8 크기의 DCT 계수를 얻을 수 있다. 축소 변환에 있어서 1/2 비율만을 제외한 나머지 비율에서 제안하는 알고리즘이 Salazar 등이 제안하는 알고리즘을 포괄할 수 없지만, 2배 또는 3배 등 정수배의 확대 변환의 경우에는 더 명확히 포함할 수 있음을 확인하였다. 또한 4장에서 제시하는 PSNR 값을 보면 이들 비율에 대해 본 논문에서 제안하는 방법의 결과 값과 Salazar 등에 의해 제안된 방법으로 실험한 결과 값이 유사함을 확인할 수 있다. 이상의 사실에서, 1/2 비율 그리고 2 또는 3배 등의 정수 비율에 있어 Salazar 등이 제안한 알고리즘은

제안하는 알고리즘의 특별한 경우로 생각할 수 있다. 따라서 Mukherjee 등이 제안한 알고리즘에서 L의 값이 1일 때의 구현 방법과 Salazar 등에 의해 제안된 방법에서 8/16 비율 즉 1/2배에 대한 구현 방법 그리고 2배 또는 3배의 정수배에 대한 구현 방법들은 본 논문에서 제안하는 알고리즘이 포괄할 수 있는 부분이다.

제안하는 알고리즘의 다른 예로 영상의 크기가 512×512에 대한 3/4 비율의 축소 변환과정을 본 논문에서 제안하는 알고리즘으로 구현하면 다음과 같다. 먼저, 제안하는 알고리즘의 과정 ①에서 3/4 비율의 축소 변환의 경우 M은 4이므로 원 영상의 8-point DCT 계수로 이루어진 블록을 4×4개 모은다. 그 다음 변환행렬  $A_{(4,8)}$ 를 이용하여 8×4인 32-point DCT 계수로 변환한다. 이것은 원 영상을 32-point DCT한 결과와 같다. 아직 영상의 전체 크기는 512×512이고, 32×32 DCT 계수 블록에서 (8×4)의 값 32와 (8×3)의 값 24의 차이인 8만큼 즉, 8×8 크기의 고주파 성분을 제거한다. 다음 과정으로, L이 3이므로 (8×L)-point IDCT 과정에서 고주파 성분이 제거된 각각의 블록에 대해 24-point IDCT를 하고 다시 8-point DCT를 함에 따라 우리는 원 영상의 크기 512×512를 3/4배로 축소 변환한 384×384 크기의 8-point DCT 계수 값으로 변환시킬 수 있다. M/L배로의 확대 변환의 경우도 마찬가지로의 방법으로 구현할 수 있다. 위의 축소 변환된 384×384 크기의 영상을 4/3배 확대 변환하여 본다. 4/3 비율이므로 M은 4, L은 3의 값을 할당하여 위의 알고리즘의 순서대로 시행하면 먼저, 384×384 크기 영상의 8×8 DCT 계수 블록을 3×3개씩 모은다. 모아진 블록을 변환행렬  $A_{(3,8)}$ 를 이용하여 24-point DCT 계수 블록으로 변환한다. 여기서 축소 변환에서는 (8×4)의 값 32가 (8×3)의 값 24보다 크므로 제거의 과정을 하였지만, 확대 변환에서는 이와는 반대로 32와 24의 차이 8만큼 0을 삽입한다. 즉 8×8 크기만큼 zero-padding함으로써 변환행렬에 의해 변환된 24×24 DCT 계수 블록은 32×32 크기의 블록으로 바뀌게 된다. 이렇게 변환된 블록을 (8×4)-point IDCT와 8-point DCT를 거쳐서 512×512 크기로 확대된 영상의 8×8 DCT 계수 블록을 얻을 수 있다.

#### IV. 실험 및 결과

DCT 영역에서 영상의 크기를 임의 비율로 변환

표 2. 1/2배 축소 변환 후 2배 확대 변환한 영상의 각각의 원 영상들과 비교

Table 2. PSNRs of various resizing methods for the 2-fold upsizing images after 1/2-fold downsizing

Algorithms	PSNR[dB]			
	lena	f-16	boats	
bilinear	30.64	29.53	29.34	
Dugad[1]	34.49	33.64	32.52	
Mukherjee[5]	35.04	34.29	32.91	
Sal.[6]	N=4, M=8	34.49	33.64	32.52
	N=5, M=10	34.92	34.21	32.85
	N=8, M=16	35.07	34.33	32.93
Park[7]	35.07	34.33	32.93	
Our	35.06	34.32	32.92	

표 3. 1/3배 축소 변환 후 3배 확대 변환한 영상의 각각의 원 영상들과 비교

Table 3. PSNRs of various resizing methods for the 3-fold upsizing images after 1/3-fold downsizing

Algorithms	PSNR[dB]			
	lena	f-16	boats	
Mukherjee[5]	31.57	29.39	29.83	
Sal.[6]	N=4, M=12	31.44	29.31	29.74
	N=6, M=18	31.55	29.39	29.81
	N=9, M=27	31.57	29.39	29.82
Park[7]	31.57	29.39	29.82	
Our	31.57	29.39	29.82	

표 4. 2/3배 축소 변환 후 3/2배 확대 변환한 영상의 각각의 원 영상들과 비교

Table 4. PSNRs of various resizing methods for the 3/2-fold upsizing images after 2/3-fold downsizing

Algorithms	PSNR[dB]			
	lena	f-16	boats	
Mukherjee [5]	37.19	37.32	35.31	
Sal.[6]	N=4, M=6	34.60	33.65	32.67
	N=6, M=9	38.40	38.64	36.11
	N=8, M=12	38.41	38.62	36.11
	N=10, M=15	38.39	38.60	36.09
Park[7]	38.41	38.62	36.11	
Our	38.72	39.09	36.21	

하는 다양한 알고리즘의 성능을 분석하기 위해서 영상을 다양한 비율로 축소한 후 다시 같은 비율로 확대 변환하여 원 영상과 비교한 PSNR 결과를 표 2~5에 각각 제시하였다. 표 2는 본 논문의 2장에서 분석했던 다양한 방법으로 원 영상들을 1/2 비율로 축소 변환시킨 후 다시 2배의 비율로 영상의 크기를 확대 변환하여 원 영상과 비교한 PSNR 값을 나타내고 있다. 이 표에서는 공간 영역에서 영상의 크기를 변환시킨 bilinear 방법에 의해 얻어낸 PSNR 값도 나타내었는데, 이 결과 값을 보면 확실히 DCT

표 5. 3/4배 축소 변환 후 4/3배 확대 변환한 영상의 각각의 원 영상들과 비교

Table 5. PSNRs of various resizing methods for the 4/3-fold upsizing images after 3/4-fold downsizing

Algorithms	PSNR[dB]			
	lena	f-16	boats	
Mukherjee [5]	38.15	38.55	36.59	
Sal.[6]	N=3, M=4	31.57	29.92	29.75
	N=6, M=8	39.73	40.38	37.68
	N=9, M=12	39.92	40.66	37.89
	N=12, M=16	39.81	40.48	37.76
Park[7]	39.92	40.66	37.89	
Our	40.21	41.13	38.20	

영역에서 영상의 크기를 변환시킨 알고리즘들의 PSNR 값보다 떨어짐을 확인할 수 있다.

이제 DCT 영역에서 영상의 크기를 변환시킨 알고리즘들의 PSNR 성능을 비교 분석한다. 먼저 1/2 배의 축소 변환 후 2배 확대 변환의 경우 Dugad 등에 의해 제안된 알고리즘을 사용하여 얻은 PSNR 결과와 Salazar 등에 의해 제안된 알고리즘에서 N은 4, M은 8로 값을 할당하여 얻어낸 PSNR 결과 값이 같음을 볼 수 있다. 이것은 Salazar 등에 의해 제안된 방법이 Dugad 등에 의해 제안된 방법을 기반으로 하여 제안한 알고리즘임을 PSNR 값으로도 확인할 수 있는 부분이다. 또한, Park 등이 제안한 알고리즘에 의해 구해진 PSNR 값과 Salazar 등이 제안한 알고리즘에서 N은 8, M은 16으로 값을 할당했을 때에 구해진 PSNR 값이 같으며, 이 접근법들에 의해 구해진 PSNR 값이 가장 높은 성능을 보인다는 사실도 확인할 수 있다. 이는 표 3~5에서 보는 것과 같이 Salazar 등이 제안한 알고리즘으로 각 비율에 맞추어 N은 9, M은 27로 할당, N은 8, M은 12로 할당 그리고 N은 9, M은 12로 각각 값을 할당했을 때 이들 모두 Park 등이 제안한 알고리즘의 결과 값과 같은 값을 얻었다. 이는 Park 등이 제안한 알고리즘이 Salazar 등에 의해 제안된 방법을 기반으로 하고 있음을 알 수 있다. 그리고 이 접근법들에 의해 구해진 PSNR 값이 가장 높은 성능을 보인다는 사실도 확인할 수 있다. 이 두 접근법이 가장 높은 성능을 가질 수 있는 근거는 원 영상을 8-point DCT하여 얻은 DCT 계수의 대부분을 유지하기 위해 IDCT와 DCT의 최적의 크기를 구하여 알고리즘을 수행하였기 때문이다. 이것은 Salazar 등에 의해 제안된 알고리즘에서 같은 비율에 대해 N과 M의 값을 달리했을 때 PSNR 값이 다르게 나타나는 것을 통해서, 원 영상의 DCT 영역에서 0의



삼입 수 또는 제거되는 고주파 성분의 DCT 계수의 크기에 의해 변환된 영상의 성능이 크게 영향을 받는다는 사실을 뒷받침한다. 영상 성능에 영향을 미치는 또 다른 요소는 변환과정 중에 사용하는 DCT 블록의 크기이다. Jiang & Feng이 제안한 DCT 계수간의 관계를 이용하여 DCT 블록의 크기를 변환할 수 있고, 여러 개의 8×8 DCT 블록을 모아서 더 큰 크기의 DCT 블록으로 변환할 수 있다.

또한, DCT 영역에서 다양한 영상의 크기 변환 방법에 대한 계산량을 다음과 같이 분석 할 수 있다. 각 방법들을 구현하기 위한 덧셈과 곱셈량을 표 6~9에 제시하였다. 먼저 Mukherjee 등이 제안한 DCT 영역에서 임의 비율 L/M로 영상의 크기를 변환하기 위한 알고리즘은 L/M 비율의 영상 크기 변환을 구현하기 위해 L배의 확대 변환 후 1/M배 축소 변환을 수행해야 한다. 따라서 계산량도 커짐을 표 6~9에서 확인할 수 있다. 예를 들어 1/2배와 3/4배로 크기를 변환할 경우의 계산량을 살펴보면, 먼저 본 논문의 2장에서 설명하였듯이 1/2배에서는 L을 1로 하여 1배 확대 변환하여야 하는데, 이것은 고려할 필요가 없고 1/2배로 축소 변환하는 과정에서 L은 2로서 그 계산량은 M을 곱셈수 A를 덧셈수라 할 때, 7M+6A이다. 따라서 Mukherjee 등이 제안한 1/2배 영상의 크기 변환의 총 계산량은 7M+6A가 된다. 하지만, 3/4배의 경우는 우선 3배의 확대 변환의 계산량은 L의 값이 3이므로 6.16M+4.72A 이고

표 6. 영상의 1/2배와 2배의 영상 크기 변환에 대한 각각의 알고리즘의 계산량  
Fig. 6. Computation amount of various resizing methods for 1/2-fold downsizing and 2-fold upsizing

Algorithms	1/2배 축소 변환		2배 확대 변환	
	곱셈수	덧셈수	곱셈수	덧셈수
Mukherjee[5]	7	6	7	5.25
Park[7]	4.96	13.03	4.96	13.03
Our	12.50	20.50	4.96	13.03

표 7. 영상의 1/3배와 3배의 영상 크기 변환에 대한 각각의 알고리즘의 계산량  
Fig. 7. Computation amount of various resizing methods for 1/3-fold downsizing and 3-fold upsizing

Algorithms	1/3배 축소 변환		3배 확대 변환	
	곱셈수	덧셈수	곱셈수	덧셈수
Mukherjee[5]	6	5.5	6.16	4.72
Park[7]	4.05	12.22	4.05	12.22
Our	11.50	20.00	4.05	12.22

표 8. 영상의 2/3배와 3/2배의 영상 크기 변환에 대한 각각의 알고리즘의 계산량  
Fig. 8. Computation amount of various resizing methods for 2/3-fold downsizing and 3/2-fold upsizing

Algorithms	2/3배 축소 변환		3/2배 확대 변환	
	곱셈수	덧셈수	곱셈수	덧셈수
Mukherjee[5]	13	10.75	13.16	10.76
Park[7]	4.24	12.85	4.24	12.85
Our	12.62	22.87	12.06	21.27

표 9. 영상의 3/4배와 4/3배의 영상 크기 변환에 대한 각각의 알고리즘의 계산량  
Fig. 9. Computation amount of various resizing methods for 3/4-fold downsizing and 4/3-fold upsizing

Algorithms	3/4배 축소 변환		4/3배 확대 변환	
	곱셈수	덧셈수	곱셈수	덧셈수
Mukherjee[5]	11.38	12.10	13.88	12.06
Park[7]	4.05	15.48	4.05	15.48
Our	11.10	25.71	12.78	23.27

그다음 1/4배의 축소 변환의 과정에서 계산량은 L의 값이 4이므로 5.22M+7.38A이다<sup>5)</sup>. 따라서 Mukherjee 등에 의해 제안된 알고리즘을 이용하여 원 영상에 대한 3/4배 크기의 영상을 구하기 위한 총 계산량은 (6.16M+4.72A)와 (5.22M+7.38A)를 합한 11.38M+12.10A가 된다. 또한, Park 등이 제안한 알고리즘은 fast DCT 알고리즘을 이용하였는데, 예를 들어 8-point DCT와 16-point DCT를 필요로 할 경우 각각 11M+29A와 31M+81A의 계산량을 필요로 한다<sup>9)</sup>. 이 알고리즘을 이용하여 1/2배 축소 변환의 경우 8-point IDCT와 16-point DCT를 수행해야하므로, 1차원의 곱셈수는 11×(2개의 블록)+31×(1개의 블록)로 그 결과는 53이고 모두 29×(2개의 블록)+81×(1개의 블록)=139의 곱셈이 필요하다. 2차원의 경우, 16번의 행계산과 8번의 열 계산이 필요하므로 1/2 배 축소 변환의 경우 (53×(8×2)+53×(8×1))/(16×16)번의 곱셈이 필요하고, 덧셈 수는 (139×(8×2))+139×(8×1))/(16×16)이므로 이것을 각각 계산하면 4.96M+13.03A가 된다. 마찬가지로 다른 실험의 계산량도 같은 방법으로 구할 수 있고, 각각의 결과 값은 표 6~9에 제시하였다.

본 논문에서 제안하는 알고리즘의 계산량은 다음과 같이 구할 수 있다. 먼저 1/2배의 축소 변환 후 2배의 확대 변환에 대한 것과 3/4배의 축소 변환 후 4/3배의 확대 변환의 계산량을 구하면 다음과 같다. 제안하는 알고리즘을 이용하여 1/2배 비율로 축소 변환하는 경우 M은 2이므로 원 영상의 8×8

DCT 블록을 2×2개 모아 변환행렬에 의해 16×16 DCT 계수 블록으로 변환하고 이로 인한 계산량은 Mukherjee 등에 의해 제안된 알고리즘의 계산량과 같으므로 7M+6A이다<sup>9]</sup>. 그리고 제안하는 알고리즘의 과정 ④에서 8-point IDCT와 8-point DCT 과정을 거치게 되는데 이로 인한 계산량은 먼저 1차원 곱의 계산은 11×1(8-point IDCT의 1개의 블록)+11×1(8-point DCT의 1개의 블록)인 22이고 덧셈의 계산량은 29×1(8-point IDCT의 1개의 블록)+29×1(8-point DCT의 1개의 블록)인 58이다. 2차원의 경우 8번의 행 계산과 8번의 열 계산이 필요하므로 곱셈 수는  $\{(22 \times 8) + (22 \times 8)\} / (8 \times 8)$ 의 결과인 5.5이고, 덧셈 수는  $\{(58 \times 8) + (58 \times 8)\} / (8 \times 8)$ 의 결과인 14.5이다. 따라서 1/2배 축소 변환의 총 계산량은 (7M+6A)와 (5.5M+14.5A)의 합인 12.5M+20.5A이다. 반대로 2배의 확대 변환은 제안하는 알고리즘의 과정 ①과 ②에서 그대로 원 영상의 8-point DCT 계수 블록을 가지므로 추가 계산이 필요하지 않다. 따라서 제안하는 알고리즘의 과정 ④에 의하여 16-point IDCT와 8-point DCT의 계산량만 구하면 되는데, 그 계산량은 Park 등이 제안한 알고리즘의 2배 확대 변환과 같으므로 그 계산량 또한 같다. 즉 16-point IDCT와 8-point DCT의 계산량으로 그 결과는 4.96M+13.03A이다<sup>7-9]</sup>.

또 다른 예로 3/4배의 축소 변환에서 제안하는 알고리즘의 과정에 따라 원 영상의 8×8 DCT 블록 4×4개를 변환행렬을 이용하여 32×32 DCT 계수 블록으로 변환할 때 계산량은 Mukherjee 등에 의해 제안된 알고리즘의 계산량과 같으므로 5.22M+7.38A이다<sup>5]</sup>. 그리고 알고리즘 과정 ④에서 24-point IDCT와 8-point DCT 과정을 거치게 되는데 이로 인한 계산량은 먼저 1차원 곱셈은 40×1(24-point IDCT의 1개의 블록)+11×3(8-point DCT의 3개의 블록)인 73이 필요하고 덧셈은 133×1(24-point IDCT의 1개의 블록)+29×3(8-point DCT의 3개의 블록)인 220이 필요하다. 2차원의 경우, 24번의 행 계산과 24번의 열 계산이 필요하므로 곱셈 수는  $\{(73 \times 24) + (73 \times 24)\} / (24 \times 24)$ 의 결과인 6.08이고, 덧셈 수는  $\{(220 \times 24) + (220 \times 24)\} / (24 \times 24)$ 의 결과인 18.33이다. 따라서 3/4배 축소 변환의 총 계산량은 (5.22M+7.38A)과 (6.08M+18.33A)의 합인 11.10M+25.71A이다. 반대로 4/3배의 확대 변환을 위해서는 원 영상의 8×8 DCT 블록 3×3개를 변환행렬에 의해 24×24 DCT 계수 블록으로 변환해야 하고 이로 인한 계산량은 Mukherjee 등이 제안한 알고리즘의 계산량과 같으므로 6M+5.5A이다<sup>5]</sup>. 그 다음 알고리

즘 과정 ④에 의하여 32-point IDCT와 8-point DCT의 계산량을 구하면 되는데, 그 계산량에서 먼저 1차원 곱은 80×1(32-point IDCT의 1개의 블록)+11×4(8-point DCT의 4개의 블록)인 124이고, 덧셈은 209×1(32-point IDCT의 1개의 블록)+29×4(8-point DCT의 4개의 블록)인 325이다. 2차원의 경우 24번의 행 계산과 32번의 열 계산이 필요하므로 곱셈 수는  $\{(124 \times 24) + (124 \times 32)\} / (32 \times 32)$ 의 결과인 6.78이고, 덧셈 수는  $\{(325 \times 24) + (325 \times 32)\} / (32 \times 32)$ 의 결과인 17.77이다. 따라서 3/4배 축소 변환을 위한 총 계산량은 (6M+5.5A)과 (6.78M+17.77A)의 합인 12.78M+23.27A이다<sup>7-9]</sup>. 나머지 비율에 대한 계산량 또한 같은 방법으로 구할 수 있고 그 결과 값들은 표 6-9에 제시하였다.

## V. 결론

DCT 영역에서 영상의 크기를 변환할 때 Dugad 등에 의해 제안된 초기의 알고리즘에서부터 Mukherjee와 Park 등에 의해 제안된 최근의 일반적인 알고리즘에 이르기까지 다양한 알고리즘들이 제안되고 있다. 본 논문에서는 DCT 영역에서 특정 비율에 대해 기존의 알고리즘을 포괄하면서, 변환된 영상의 품질을 향상시킬 수 있는 임의 영상 크기 변환 방법을 제안하였다. 따라서 영상 크기 변환에 가장 좋은 성능을 보이는 Mukherjee 등에 의해 제안된 알고리즘과 Park 등에 의해 제안된 알고리즘을 적용하였다. 영상의 성능을 높이기 위해서 영상의 화소간 상관도를 최대한으로 이용하기 위해 원 영상의  $(8 \times M) \times (8 \times M)$  크기의 DCT 계수 블록으로 변환시키는 변환행렬을 이용하는 방법과 변환하고자 하는 비율 값에 맞추어 IDCT 크기를 결정하는 방법을 사용하였다, 따라서 본 논문은 영상의 성능을 높이기 위해 위의 두 요소를 모두 고려한 알고리즘을 제안함으로써 영상의 크기변환 시에 향상된 성능을 보이는 것을 확인할 수 있었다. 그리고 특정 비율에 있어서 기존의 알고리즘을 포괄할 수 있는 모습도 확인할 수 있었다.

## 참 고 문 헌

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

[2] Q.,Hu, and S., Panchanathan “Image/video spatial scalability in compressed domain”, *IEEE Trans, Und. Electron.*, vol. 45, no. 1, pp. 23-31, Feb., 1998.

[3] Mukherjee, J., and Mitra, S.,K. “Image resizing in the compressed domain using subband DCT”, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 7, pp. 620-627, July 2002.

[4] Jiang, J., and Feng, G. “The spatial relationship of dct coefficients between a block and its subblocks”, *IEEE Trans. Signal Process.*, vol. 50 , no. 5, pp. 1160-1169, May, 2002.

[5] Mukherjee, J., and Mitra, S.,K. “Arbitrary re-sizing of images in DCT space”, *IEEE Proc.-Vis. Image Signal Process.*, vol. 152, no. 2, pp. 155-164, April, 2005.

[6] C. L. Salazar and T. D. Tran, “On resizing images in the dct domain”, in proc. *IEEE Int. Conf. Image Processing*, Singapore, pp. 2797-280 0, Oct., 2004.

[7] Y. Park and H. Park, “Arbitrary-ratio image resizing using fast dct of composite length for dct-based transcoder”, *IEEE Trans. Image Processing*, vol. 15, no. 2, pp. 494-5 00, Feb., 2006.

[8] 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.

[9] 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, pp. 988-991 ,May, 1989.

김 용 재 (Yong-Jae Kim)

준회원



2006년 8월 가톨릭대학교 컴퓨터 전자공학부 학사 졸업  
2006년 8월~현재 가톨릭 대학교 컴퓨터공학과 석사과정  
<관심분야> 신호처리, 영상통신, 영상처리

이 창 우 (Chang-Woo Lee)

정회원

한국통신학회 논문지 제31권 제4C호 참조  
<관심분야> 영상통신, 영상처리