

# 밝기 성분을 기준으로 한 프랙탈 컬러 영상 압축

오 정 석<sup>†</sup> · 안 재 형<sup>††</sup>

## 요 약

본 논문에서는 프랙탈 변환을 이용하여 컬러 영상을 압축한다. 컬러 영상을 RGB 공간에서 YCbCr 컬러 공간으로 변환하여 Y 성분에 프랙탈 변환을 적용시키고, Cb와 Cr 성분은 scaling과 offset 값을 변화시켜 블록 탐색에 소비되는 시간을 기존보다 약 1/3로 줄일 수 있다. 또한 성능 향상을 위해 Quad-tree 가변 블록 분할 방법을 이용하였다. 모의 실험 결과 기존 방법에 비하여 제안 알고리즘이 고정 블록 분할 방법에서는 약 1.5~2.2 [dB] 정도, Quad-tree 블록 분할 방법의 경우 약 2.1[dB]의 성능 향상을 보였다.

## A Fractal Color Image Compression Technique based on the Luminance Component

Jeong Seok Oh<sup>†</sup> · Jae Hyoung Ahn<sup>††</sup>

### ABSTRACT

This paper proposes a fractal color image compression technique based on the luminance component. It uses the fractal parameters computed from the luminance component as the basis values of the fractal parameters for the other color components. The adaption of scaling and offset values for the color components can reduce the searching time to one-third rate. Simulation results show that the proposed technique provides more 1.5~2.2[dB] in the fixed block-searching method and about 2.1[dB] in the Quad-tree block-searching method than the conventional fractal coding technique.

### 1. 서 론

프랙탈 영상 압축은 1992년 Jacquin<sup>[1]</sup>이 디지털 영상에 대해 컴퓨터를 이용한 압축 알고리즘을 제시한 이후 많은 사람들에 의해 연구되어져 왔다. 프랙탈 압축의 기본적인 원리는 영상내의 자기 유사성을 이용해 원래 영상을 압축 및 복원할 수 있다는 것이다. 압축 방법은 먼저 영상을 지역 블록과 이보다 큰 정의역 블록으로 나눈다. 그리고 각각의 지역 블록에

대하여 블록의 변환을 통해 모든 정의역 블록과의 비교로 가장 적합한 블록을 찾는다. 마지막으로 이러한 변환 계수들을 저장함으로써 영상을 압축하게 된다. 복원 과정은 이 변환 계수들의 반복적인 적용을 통해 원래의 영상을 얻게 된다.

컬러 영상에 있어서 이러한 프랙탈 영상 압축 방법을 적용하는 가장 기본적인 방법으로는 각각의 컬러 성분에 대해 별도의 압축을 하는 것이다. 그러나 이 방법은 각 컬러 성분간의 상관 관계를 전혀 고려하지 않은 것으로 압축 효율이 낮다. 보다 큰 압축율을 얻기 위해서 Zhang과 Po<sup>[2]</sup>는 벡터 왜곡 척도(vector distortion measure)를 사용하였다. 이 방법은 블록 탐색

<sup>†</sup> 준 회 원:충북대학교 공과대학 정보통신공학과  
<sup>††</sup> 정 회 원:충북대학교 공과대학 정보통신공학과  
 논문접수:1997년 9월 23일, 심사완료:1997년 12월 4일

에 있어서 각각의 픽셀에 대해 벡터로 RGB 값들의 거리(distance)를 비교한다. 이렇게 함으로써 벡터간 거리가 가장 가까운 블록을 탐색하고, 별도의 컬러 성분으로 압축하는 것 보다 향상된 성능을 얻었다. 그러나 프랙탈 영상 압축에 소비되는 대부분의 시간이 블록 탐색에 있는 점을 고려할 때 이 방법은 계산량이 매우 크다는 단점을 갖는다.

본 논문에서는 이러한 문제를 해결하기 위해 새로운 알고리즘을 제안한다. 제안 알고리즘은 먼저 RGB 컬러 공간을 YCbCr 컬러 공간으로 변환한다. Y 성분에 대해 프랙탈 압축 기법을 적용하여 충분한 복원이 이루어지도록 하고, Cb와 Cr 성분은 일부의 코드 값을 공유함으로써 성능의 향상과 계산 속도의 향상을 얻고자 한다. 서론에 이어 II장에서 RGB 컬러 성분에 의한 프랙탈 영상 압축을 살펴보고 III장에서는 YCbCr 컬러 공간을 이용하여 효율적인 압축 알고리즘을 제안한다. IV장에서 컴퓨터 모의실험을 통한 결과를 이용해 기존 알고리즘과 제안 알고리즘을 비교하고, V장에서 결론을 내렸다.

## 2. RGB 컬러 공간에서의 벡터 왜곡 척도를 이용한 컬러 영상 압축

디지털 영상에 대한 프랙탈 압축 방법의 간단한 예는 다음과 같다. 먼저  $N \times N$  영상을  $n \times n$ 의 치역 블록으로 서로 중첩되지 않도록 분할한다. 그리고 동일한 영상에 대해 치역 블록의 두배 크기인  $2n \times 2n$ 으로 구성된 정의역 블록을 서로 50% 중첩하여 분할한다. 각각의 정의역 블록은 일반적으로 4개 픽셀의 평균값을 이용해 치역 블록과 같은 크기로 서브 샘플링(sub-sampling) 된다. 다음으로 식 (1)의 블록간 거리와 식 (2)의 유사변환(affine transform)을 이용해 가장 적절하게 매핑(mapping)되는 블록을 탐색하게 된다.

$$d(R, R') = \sqrt{\frac{1}{n^2} \sum_{(x,y) \in R, R'} [f(x,y) - f'(x,y)]^2} \quad (1)$$

$R: n \times n$  크기의 정의역 블록

$R': n \times n$  크기의 치역 블록

$f(x, y), f'(x, y): R, R'$  블록의  $x, y$ 좌표에서 픽셀의 grayscale 값

$$w \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i \end{bmatrix} \quad (2)$$

식 (1)은  $n \times n$  픽셀로 되어있는 두 개의 블록  $R$ 과  $R'$ 간의 닮음 정도를 표시하며 식 (2)는 탐색된 블록의 위치, 회전, 대칭, 계조도의 변화 등을 나타낸다. 여기서 계수  $a_i, b_i, c_i, d_i, e_i, f_i, s_i$ , 그리고  $o_i$ 는 모두 프랙탈 코드를 만드는 변수이며 이중에  $s_i$ 는 scaling 값이고  $o_i$ 는 offset 값이 된다. 탐색된 블록에 대해 각각의 계수를 구해 이 값을 저장함으로써 압축을 하게 된다.

복원 과정은 이렇게 얻어진 계수들을 이용해 임의의 초기 영상에 대해 반복적으로 동일한 변환을 적용하여 원래의 영상을 얻게 된다. 식 (3)은 Collage 정리로 원영상  $f_{org}$ 와 변환영상  $w(f_{org})$ 의 차이가 적을수록 임의의 영상  $f_0$ 의  $n$ 번 반복된 수축 변환(contractive transform)이 원영상에 근사화 되어진다.

$$d(f_{org}, w^n(f_0)) \leq \frac{1}{(1-s)} \cdot d(f_{org}, w(f_{org})) \quad (3)$$

Jacquin<sup>[3]</sup>은 코딩의 효율을 높이기 위해 블록 분류 방법을 이용했으며, 2단계(bi-level) 탐색을 이용해 서서히 변화하는 영역에 대해 큰 블록을 사용하고 복잡한 영역에 대해서는 작은 블록을 사용하여 영상을 압축했다. Fisher<sup>[4][5]</sup>에 의해 프랙탈 코딩에 이용된 Quad-tree 블록 분할 방법은 가변 블록 분할 방식이다. 이 방법은 우선 일정한 크기의 치역 블록에서 치역 블록 크기의 두배인 정의역 블록들을 탐색한다. 이때 미리 정해진 에러 임계값(error threshold)을 만족하는 블록이 없으면 치역 블록을 4개의 하위 블록으로 나눈다. 각 블록들에 대해 다시 두배 크기의 정의역 블록을 탐색하여 에러 임계값 이하의 블록을 찾거나, 최소 블록 크기가 될 때까지 이것을 되풀이하는 방법이다. 이 방법은 고정 블록 분할 방법에 비해 블록을 적용적으로 분할할 수 있으므로 큰 블록을 사용할 수 있는 영역에 대해서는 블록을 크게 나눔으로써 압축율을 높이고, 작은 블록으로 분할된 부분들은 좀더 영상을 세밀히 표현할 수 있다.

컬러 영상에 대한 압축 방법은 흑백 영상에 적용된 압축 방법을 각각의 컬러 성분에 그대로 적용할 수

있다. 그러나 각각의 컬러 성분에 대해 별도의 코딩을 하는 것은 컬러 성분 상호간의 상관관계를 적절히 이용하지 못하게 된다. 이러한 문제를 해결하기 위해 Zhang과 Po<sup>[2]</sup>는 블록의 비교에 식 (4)와 같은 벡터 왜곡 척도를 사용하였다. 또한 유사변환에 3차 맵(3-dimensional map) 대신에 식 (5)의 5차 맵(5-dimensional map)을 이용 하였다.

$$f(x, y) = [r, g, b]^T,$$

$$d(R, R') = \sqrt{\frac{1}{3m^2} \sum_{(x, y) \in R, R'} |f(x, y) - f'(x, y)|^T |f(x, y) - f'(x, y)|} \quad (4)$$

$r, g, b$ : 각 컬러 성분의 값

$$w \begin{bmatrix} x \\ y \\ r \\ g \\ b \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 & 0 & 0 \\ c_i & d_i & 0 & 0 & 0 \\ 0 & 0 & s_i^r & 0 & 0 \\ 0 & 0 & 0 & s_i^g & 0 \\ 0 & 0 & 0 & 0 & s_i^b \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ r \\ g \\ b \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i^r \\ o_i^g \\ o_i^b \end{bmatrix} \quad (5)$$

Zhang과 Po는 R, G, B 컬러 성분이 각각 별도의 256 level을 갖는 gray 영상으로 볼 수 있기 때문에, 식 (4)의 벡터 왜곡 척도를 이용하여 벡터 공간에서 R, G, B를 축으로 한 값들의 거리를 비교하였다. 즉, 정의역 블록과 치역 블록간에 가장 유사한 블록을 탐색하기 위하여 블록내 각각의 R, G, B 값을 3차원 벡터 공간의 한 점으로 하여 이와 가장 가까운 거리에 있는 블록을 탐색하게 된다. 이렇게 얻어진 블록에 대하여 좌표값과 isometry 변환값인 식 (5)의 a, b, c, d, e, f 변수의 값을 얻고, R, G, B 각각의 값에 대하여 적절한 scaling과 offset값을 계산하여 다른 변수값을 구하게 된다. 이 방법은 각각의 컬러 성분을 인코딩할 때 벡터적으로 가장 가까운 블록을 탐색하기 때문에 별도의 컬러 성분으로 압축하는 것에 비하여 PSNR의 손실을 가져오지만, 압축에 필요한 좌표값과 isometry 변환값을 한번만 저장하므로 보다 높은 압축율을 얻을 수 있다.

Zhang과 Po의 실험 결과를 보면 각각의 컬러 성분을 별도로 압축한 방법에서 압축율 30.12일 때 27.33 [dB]의 PSNR을 나타내지만, 벡터 왜곡 척도를 이용

했을 경우 압축율 46.54에서 26.90[dB], 압축율 43.88에서 27.46[dB]의 PSNR을 얻어 각각의 컬러 성분에 대해 별도의 코딩을 이용하는 것보다 약 1.5배의 압축율 향상을 얻었다.<sup>[2]</sup> 그러나 이 알고리즘은 벡터 공간에서의 계산을 요구하므로 블록탐색에 필요한 시간을 줄이는데는 효율적이지 못하다.

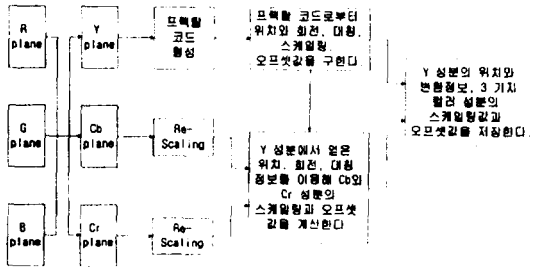
### 3. 컬러 영상에서의 효율적인 블록 탐색

블록의 탐색 과정은 프랙탈 영상 압축에서 가장 많은 시간을 소비하는 곳이다. 일반적으로 흑백 영상의 블록 탐색에는 식 (1)을 이용해 각 블록간의 거리를 측정하게 된다. 이 식을 살펴보면 블록의 크기가 8×8일 때 한 개의 블록을 비교하는데 적어도 2<sup>3</sup>×2<sup>3</sup>의 곱셈 계산이 필요하다. 컬러 영상에 대하여 Zhang과 Po가 제안한 방법을 살펴보면 한 블록에 대한 비교시 곱셈의 횟수는 2<sup>3</sup>×2<sup>3</sup>×3이 된다. 따라서 각각의 컬러 성분에 대하여 프랙탈 영상 압축 방법을 적용하는 것과 계산량의 차이는 별로 없게 된다. 이러한 점을 살펴볼 때 기존의 알고리즘은 컬러 영상에 대한 압축율을 높일 수는 있으나 블록 탐색에 소비되는 시간에 대해서는 효과적이지 못하다.

일반적으로 컬러 영상 압축에는 RGB 컬러 성분을 휘도(luminance)와 색차(chrominance)신호로 분리하기 위한 변환이 사용된다. 인간의 시각 체계가 색차(chromatic)신호의 변화보다 밝기(achromatic)신호의 변화에 훨씬 민감하기 때문이다. 이러한 특성을 이용하여 JPEG 영상 압축에서는 컬러에 관계없이 각각의 R, G, B 성분별로 코딩을 하지만 휘도와 색차신호로 분리하여 색차신호는 서브샘플링(Subsampling)을 통해 영상의 크기를 줄여 압축하는 방법을 사용한다.<sup>[6]</sup> 이렇게 RGB 신호를 휘도와 색차신호로 변환하는 방법은 YIQ, YUV, 그리고 YCbCr등의 여러 가지 방법이 있다. YCbCr은 식 (6)과 같이 YUV의 등가 식으로 아날로그 신호가 아닌 디지털 신호를 위한 변환식이며 일반적인 비디오 신호는 이 신호형태로 저장된다.<sup>[7][8]</sup>

$$\begin{aligned} Y &= 0.299 \times R + 0.587 \times G + 0.114 \times B \\ Cb &= -0.169 \times R - 0.331 \times G + 0.500 \times B \\ Cr &= 0.500 \times R - 0.419 \times G + 0.081 \times B \end{aligned} \quad (6)$$

본 논문에서는 Zhang과 Po가 제안한 알고리즘의 단점인 계산속도의 향상과 프랙탈 영상 압축 방법을 고려하여, 컬러 공간의 변환을 이용해 (그림 1)과 같은 알고리즘을 제안한다.



(그림 1) 제안 알고리즘의 부호화 블록도

(Fig. 1) Blockdiagram of proposed encoding algorithm

제안 알고리즘은 우선 RGB 컬러 공간을 YCbCr 컬러 공간으로 변환한다. 다음으로 영상의 형태에 대한 대부분의 정보를 갖고 있는 Y 성분을 이용해 가장 유사한 블록을 탐색한다. 이때 얻어진 좌표와 isometry 변환값을 이용해 색차신호인 Cb와 Cr 컬러 성분은 단지 scaling과 offset 값을 재계산한다. 이와 같이 각 컬러 성분의 특성을 이용해 코딩함으로써 보다 나은 효율을 얻도록 하였다. 식 (6)에서 Cb와 Cr 성분은 음의 값에서 양의 값까지 범위를 갖기 때문에 Y 성분에 적용된 것과 동일한 알고리즘을 적용하기가 어렵다. 이러한 문제점을 해결하기 위하여 값의 범위를 0~255 까지로 재배열(re-scaling) 한다. 즉, RGB 컬러 공간을 YCbCr 컬러 공간으로 변환하는 과정에서 Cb와 Cr의 최대값과 최소값 사이를 0~255 까지로 재배열한다. 이때 16 바이트의 부가정보를 저장하게 된다.

제안 방법은 저장할 정보의 양은 기존의 알고리즘과 비슷한 값을 갖게되며 블록 탐색을 위한 계산은 감소하게 된다. 예를 들어 512×512 영상에 대해 치역 블록의 크기를 8×8로, 정의역 블록을 16×16으로 중첩없이 분할할 경우 Zhang과 Po가 제안한 방법과 비교하면 <표 1>과 같다. <표 1>에서 보는 바와 같이 영상의 블록 탐색에 필요한 계산량에 비하면 컬러 공간의 변환에 필요한 계산량은 얼마 되지 않는다. 즉 기존 알고리즘에 비해 블록탐색에 소비되는 시간은 1/3 정도밖에 되지 않는다.

<표 1> 블록 탐색을 위한 곱셈의 계산량 비교  
<Table 1> The number of multiplication to search proper block

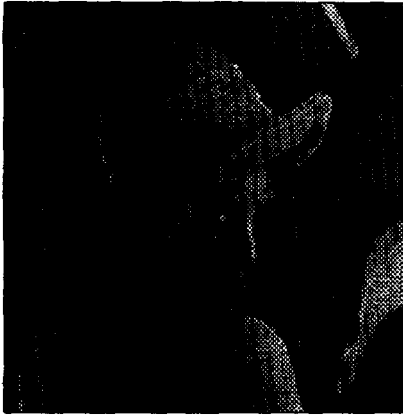
	Zhang&Po 알고리즘	제안 알고리즘
치역 블록수	$2^6 \times 2^6$	$2^6 \times 2^6$
정의역 블록수	$2^5 \times 2^5$	$2^5 \times 2^5$
isometry 변환	$2^3$	$2^3$
한 개 블록을 비교	$2^3 \times 2^3 \times 3$	$2^3 \times 2^3$
전체 영상에 대해 블록 탐색에 필요한 곱셈수	$3 \times 2^{31}$	$2^{31}$
Rescaling을 위한 전처리	-	$(2^9 \times 2^9) \times (3 \times 2)$
RGB ↔ YCbCr (나눗셈 포함)	-	$2 \times (2^9 \times 2^9) \times (3 \times 3 + 2 \times 2)$
추가된 곱셈수		$2^{23}$
전체 곱셈수의 차이		$3 \times 2^{31} - (2^{31} + 2^{23}) \approx 2^{32}$

일반적으로 프랙탈 영상 압축에서 성능을 향상시키기 위해 가변 블록 분할 방법을 사용한다. 본 알고리즘에서의 블록 분할 방법은 Quad-tree 분할 방법을 사용한다. 이 방법은 가변 블록 분할 방법이므로 고정 블록 분할 방법보다 전체적인 영상의 화질과 부분적인 영상의 화질 모두를 개선할 수 있다.

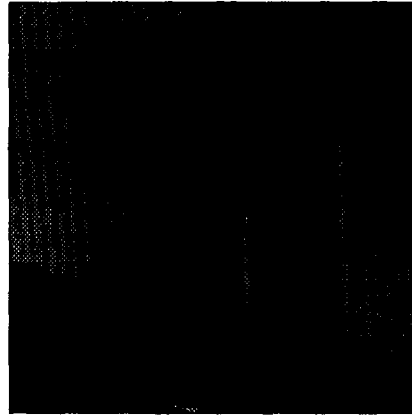
압축된 파일로부터의 복원과정은 (그림 1)에서 압축 순서의 역 과정으로 하나의 압축 파일로부터 좌표, 회전, 그리고 대칭정보와 각 컬러 성분의 스케일링과 오프셋값을 읽는다. 이 값들을 이용해 세 개의 새로운 영상에 반복 적용하여 원래의 영상을 얻는다. 마지막으로 복원된 YCbCr 컬러 공간을 RGB 컬러 공간으로 변환하게 된다.

#### 4. 모의실험 결과 및 고찰

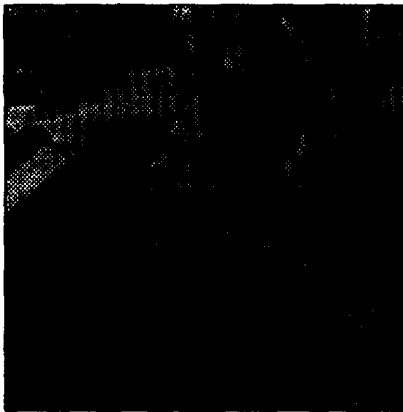
본 논문에서는 512×512 크기의 컬러 영상 Lena에 대해서 고정 블록 분할 방법을 적용하여 기존의 방법과 비교해보고, 또한 Bridge, Barbara 그리고 Mandrill 영상에 대해서도 제안한 방법에 의한 모의실험을 수행하였다. 가변 블록 분할 방법인 Quad-tree 방법을 사용하였으며 객관적인 성능 평가의 척도로 식 (7)과 같이 원영상과 복원영상의 MSE에 대한 평균값을 이용한 PSNR(Peak Signal to Noise Ratio)을 사용하였다.



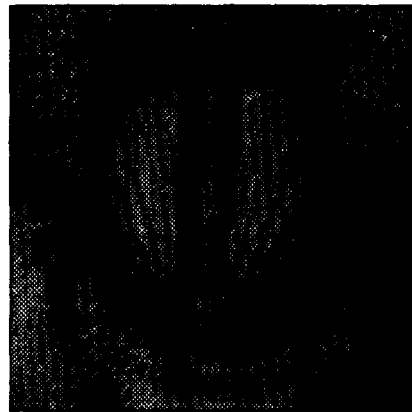
(a) Lena



(b) Bridge



(c) Barbara



(d) Mandrill

(그림 2) 원영상 (512×512, 컬러영상)  
(Fig. 2) Original images (512×512, color image)

$$PSNR = 10 \log \frac{255^2}{MSE} \quad [dB] \quad (7)$$

〈표 2〉를 살펴보면 Lena 영상에 있어서 기존의 RGB 컬러 공간에서 벡터 왜곡 척도를 사용한 경우 압축율 43.88에서 27.46[dB]의 PSNR을 얻는 반면, 제안한 방법을 이용한 경우 압축율 42.62에서 29.39[dB]를 얻었다. 또한 기존방법에서 압축율 10.97에서 31.11[dB]의 PSNR을 얻는 반면, 제안 방법에서는 압축율 10.11에서 33.16[dB]를 얻었다.

(그림 3)은 압축율과 PSNR을 비교해 나타낸 것으로 기존의 방법에 비하여 제안한 방법이 약 1.5~2.2[dB] 정도 성능이 향상되었음을 볼 수 있다. RGB 컬러 공간에 벡터 왜곡 척도를 이용한 방법에 비하여 제안 방법이 계산량이 1/3 정도임에도 불구하고 보다 향상된 성능을 얻을 수 있음을 알 수 있다.

또한 3 단계 Quad-tree 블록 분할 방법을 이용하여 실험 영상에 적용한 결과를 (그림 4)에 보였다. 그림에서 보는 바와 같이 Lena 영상에 대하여는 Zhang과 Po가 얻었던 압축율 27.79에서 29.69[dB]보다 더욱 향

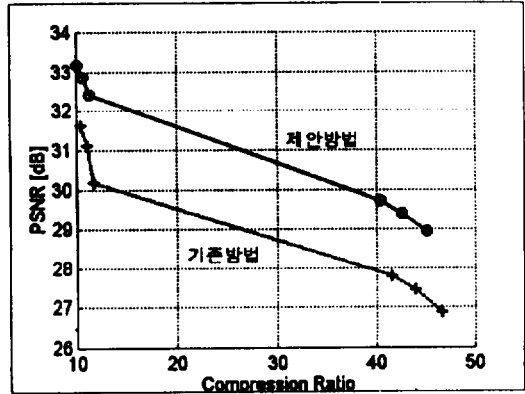
<표 2> Lena 영상에 대하여 기존방법과 제안방법을 이용한 프랙탈 컬러 영상 부호화의 성능

<Table 2> Performance of fractal color image coding between traditional and proposed algorithm

	지역블록	정의역블록	$\Delta h, \Delta v$	compression ratio	PSNR(dB)
기존방법	8×8	16×16	16	46.54	26.90
			8	43.88	27.46
			4	41.51	27.80
	4×4	8×8	16	11.64	30.17
			8	10.97	31.11
			4	10.38	31.64
제안방법	8×8	16×16	16	45.14	28.93
			8	42.62	29.39
			4	40.38	29.71
	4×4	8×8	16	11.29	32.41
			8	10.66	32.84
			4	10.10	33.16

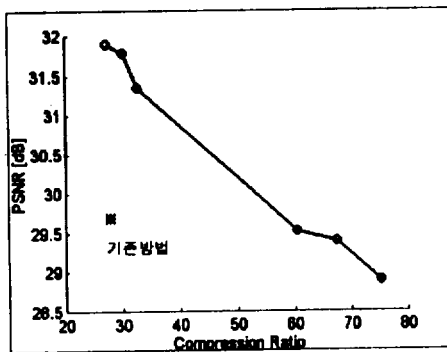
$\Delta h, \Delta v$ 는 각각 수평 수직 방향의 step 크기

상된 압축율 30.15에서 31.78[db]를 얻을 수 있었다. (그림 5)는 제안 방법으로 영상을 압축한 후 복원된 영상이다. Bridge에서 약간의 artifact가 발생하는데

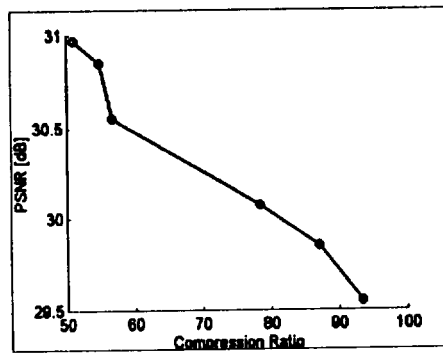


(그림 3) Lena 영상에 대한 기존방법과 제안방법 간의 압축율 대 PSNR

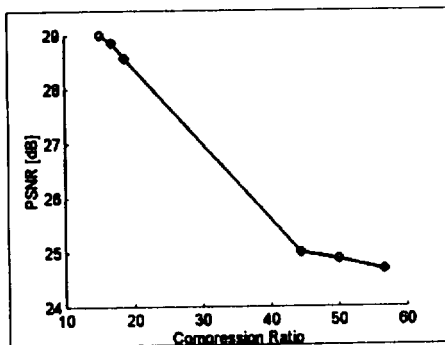
(Fig. 3) Compression ratio vs. PSNR between conventional and proposed method



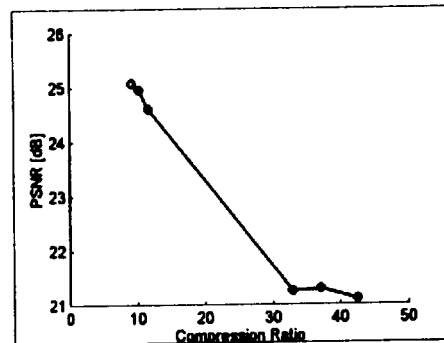
(a) Lena



(b) Bridge



(c) Barbara



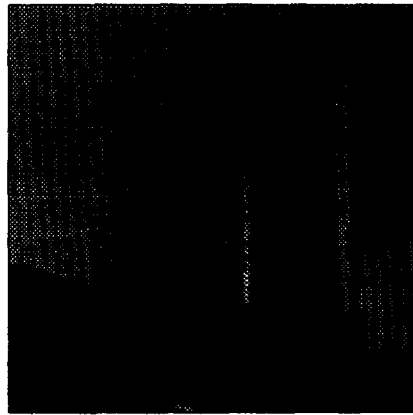
(d) Mandrill

(그림 4) 모의 실험 결과

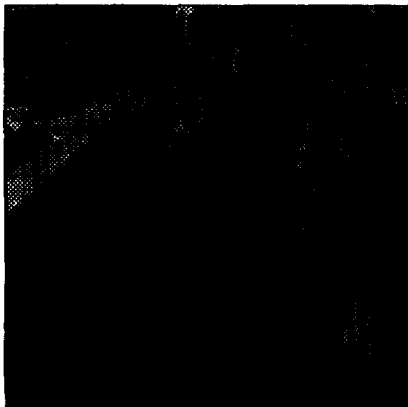
(Fig. 4) Simulation results



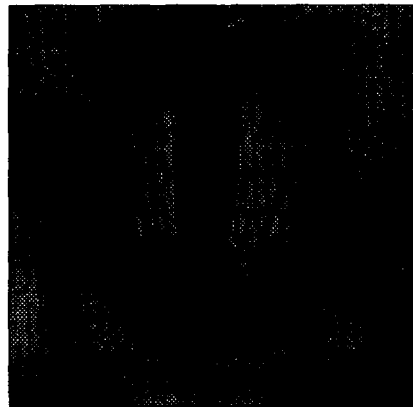
(a) Lena(압축율 : 30.15 PSNR : 31.78)



(b) Bridge(압축율 : 54.86 PSNR : 30.85)



(c) Barbara(압축율 : 16.90 PSNR : 28.86)



(d) Mandrill(압축율 : 10.27, PSNR : 24.96)

(그림 5) 복원 영상  
(Fig. 5) Reconstruction images

이것은 높은 압축율을 얻기 위하여 블록의 크기가 커짐으로서 발생하며, 보다 작은 블록으로 분할한다면 보다 원 영상에 가까운 영상을 얻을 수 있다.

## 5. 결 론

프랙탈 영상 압축 알고리즘을 컬러 영상에 적용하는데 있어서 기존의 RGB 컬러 공간에 벡터 왜곡 척도를 이용한 방법은 각 컬러 성분을 별도로 압축하는 것과 비교해 계산량에서의 이득은 얻을 수 없었다. 이

러한 단점을 개선하기 위하여 본 논문에서는 YCbCr 컬러 공간으로 변환하여 Y 성분을 압축하고 나머지 Cb와 Cr 성분은 코드의 일부를 공유함으로써 계산량을 줄이는 방법을 제안하였다. RGB 컬러 공간을 YCbCr 컬러 공간으로 변환하면 영상의 밝기와 형태에 관한 대부분의 정보가 Y 성분에 들어 있게 된다. 이러한 점을 이용하여 Y 성분에 대하여 프랙탈 압축을 행한 후 나머지 성분들에 대해서는 단지 scaling과 offset 값을 변화시킴으로써, 블록 탐색에 소비되는 계산량을 1/3정도 감소시킬 수 있었다. 또한 영상의

화질에 있어서도 제안 방법이 기존의 방법에 비하여 1.5~2.2[dB] 정도 향상되었다.

제안된 알고리즘은 블록의 크기를 영상의 내용에 따라 가변적으로 분할하는 Quad-tree 분할 방법을 사용하였다. 이러한 Quad-tree 분할 방법의 사용에 있어서 Cb와 Cr성분의 scaling과 offset에 할당되는 비트율을 적절히 조절하거나, Y 성분에서 탐색된 레벨의 상위 레벨에서 코딩을 한다면 보다 큰 압축율을 얻을 수 있을 것으로 기대된다.

### 참 고 문 헌

- [1] A. E. Jacquin, "Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations", IEEE Transactions on Image Processing, vol. 1, pp. 18-30, Jan. 1992.
- [2] Y. Zhang and L. Po, "Fractal Color Image Compression Using Vector Distortion Measure", Proc. ICIP '95 IEEE International Conference on Image Processing, pp. 276-279, 1995.
- [3] A. E. Jacquin, "Fractal Image Coding: A Review", Proceedings of the IEEE, vol. 81, pp. 1451-1465, Oct. 1993.
- [4] Y. Fisher, "Fractal Image Compression", SIG-GRAPH '92 Course Notes, 1992.
- [5] Y. Fisher (Editor), Fractal Image Compression-Theory and Application, Springer-Verlag, NewYork, 1994.
- [6] M. Rabbani and P. W. Jones, Digital Image Compression Techniques, SPIE Optical Engineering Press, 1991.
- [7] Vasudev Bhaskaran and Konstantinos Konstantinides, Image and Video Compression Standards, Kluwer Academic Publishers, 1995.
- [8] Andy C. Hung, MPEG, CCITT H.261(P\*64), JPEG, Portable Video Research Group Stanford University, 1993.



### 오 정 석

1996년 2월 충북대학교 정보통신공학과 졸업(공학사)  
 1996년 3월~현재 충북대학교 대학원 정보통신공학과 재학  
 관심분야: 영상신호처리, 영상압축



### 안 재 형

1981년 2월 충북대학교 전기공학과 졸업(공학사)  
 1983년 2월 한국과학기술원 전기 및 전자공학과(공학석사)  
 1992년 2월 한국과학기술원 전기 및 전자공학과(공학박사)  
 1987년~현재 충북대학교 정보통신공학과 교수  
 관심분야: 영상통신 및 데이터 감축, 인간의 시각 효과 및 응용