

## 새로운 코드북을 이용한 반복 적용이 필요없는 프랙탈 영상 부호화 기법

正會員 金昌洙\*, 白勳\*, 李商郁\*

## A Non-iterative Fractal Image Coding Technique Using Novel Codebooks

Chang Su Kim\*, Hoon Paek\*, Sang Uk Lee\* Regular Members

## 要 約

본 논문에서는 복호기에서 반복 적용이 필요없는 새로운 프랙탈 부호화 기법을 제안한다. 기존에 제안된 반복 적용이 필요없는 프랙탈 부호화기[12]는 레인지 블록의 크기가  $8 \times 8$  이하로 제한되므로, 화소값이 부드럽게 변화하는 영역을 효율적으로 부호화하지 못하는 단점이 있다. 본 논문에서는 이 문제점을 완화하기 위하여, 평면 근사된 영상에서 두개의 코드북을 발생시킨다. 첫째 코드북은 화소값이 부드럽게 변화하는 큰 레인지 블록을 부호화하기 위하여 스무딩 변환을 이용하여 발생시키고, 둘째 코드북은 화소값이 급격히 변화하는 작은 레인지 블록을 부호화하기 위하여 공간 축소 변환을 이용하여 발생시킨다. 실제 영상에 대한 컴퓨터 모의 실험을 통하여 제안하는 알고리즘이 다른 프랙탈 부호화 알고리즘에 비하여 객관적인 화질과 주관적인 화질의 측면에서 성능이 향상됨을 보인다. 또한, 제한하는 알고리즘은 복호기에서 반복 적용이 필요없으므로 복호화 시간이 매우 빠르다는 장점이 있다.

## ABSTRACT

In this paper, we propose a novel fractal image compression technique, which does not require iteration at the decoder. The main problem relating to the conventional non-iterative algorithm[12] is that the smooth region cannot be coded efficiently, since the size of the range block is limited to be less than  $8 \times 8$ . We alleviate this problem by generating two codebooks from planarly approximated image. In other words, the first codebook is generated by the smoothing operator for large and smooth range blocks and the second codebook is generated by the spatial contraction operator for small and active range blocks, respectively. The computer simulation results on the real images demonstrate that the proposed algorithm provides much better performance than most other fractal-based coders, in terms of the subjective quality as well as the objective quality (PSNR). Moreover, the proposed algorithm is very fast in decoding, since it does not require iteration at the decoder.

\* 서울대학교 工科大学 制御計測工學科  
論文番號 : 95195-0530  
接受日字 : 1995年 5月 30日

## 1. 서 론

영상 데이터는 그 정보량이 매우 방대하기 때문에 제한된 전송 채널이나 저장 매체에서 효율적인 활용을 위하여 정보량의 감축이 반드시 필요하다. 따라서, 많은 영상 압축 기법이 제안되어 왔는데, 변환 부호화 기법이나 벡터 양자화기(VQ)를 이용한 기법등이 그 예이다<sup>[1,2]</sup>. 이러한 영상 압축 기법들 중에서 여러가지 표준안들이 제시되고 있는데 정지영상의 경우에는 JPEG, 동영상의 경우에는 H.261, MPEG-I 그리고 MPEG-II 등이 제안되어 있다<sup>[3,4,5,6]</sup>. 하지만 기존의 방법을 이용하여 영상의 압축비를 매우 높게 하면 심각한 화질의 저하가 발생하므로, 아주 낮은 전송율을 가지는 영상 압축 기법이 현재 활발히 연구되고 있다<sup>[7]</sup>. 이런 초저전송율 영상 압축 부호화(very low bit-rate coding)를 위하여 기존의 방법과는 전혀 다른 접근 방법으로서 최근 프랙탈 영상 압축 기법이 주목을 받고 있다.

프랙탈 기하학은 나무나 구름같이 복잡한 자연 물체를 표현하기에 적합한 기하학이다<sup>[8]</sup>. 유클리드 기하학이 직선이나 원, 타원 등의 객체를 간단히 표현할 수 있어 인공적인 물체를 표현하기에 적합하지만, 자연 물체를 표현하기에는 부적합하여 프랙탈 기하학이 개발되었다. 자연 물체에는 자기 유사성이 존재하는데, 자기 유사성이란 나무의 예에서 전체 나무와 그 나무에 있는 하나의 가지가 서로 유사하게 생긴 특성을 의미한다. 프랙탈은 이 자기 유사성을 이용하여 간단한 함수를 가지고 무한의 해상도를 가지는 영상을 표현하는데, Barnsley는 이를 처음으로 영상 압축 기법에 적용하고 그 이론적 기반을 세웠다<sup>[9]</sup>. 즉, 영상 공간(image space)에서 부호화하고자 하는 영상을 고정점(fixed point)으로 가지는 축소 변환 함수(contraction map)를 찾고, 복호기에서는 이 함수를 임의의 영상에 반복 적용하면 원 영상을 얻을 수 있다. 만약 이 함수 정보가 원 영상보다 적은 비트로 표현 가능하다면 부호화 이득을 얻게 된다. Barnsley는 특정한 영상에 대하여 매우 높은 압축율(10000 : 1)을 가지는 결과를 발표하였는데, 축소 변환 함수를 찾는 방법이 완전히 자동화된 알고리즘이 아니어서 입력 영상에 따라 사람이 수작업으로 부호화하여야 했다. 이 부호화 문제를 해결하여 Jacquin이 블리블로 축소 변환 함수를 찾는 자동화된 알고리즘을 처음 제시한 후에 프랙탈 영상 압축 기법을 위한 많은 연구가 진

행되고 있다<sup>[10]</sup>. Jacquin의 알고리즘은 부호화 방법이 너무 복잡하여 부호화 시간이 오래 걸린다는 것과 복호화시 반복 적용이 필요하다는 단점이 있는데, 이를 개선하기 위하여 많은 방법이 시도되어 왔다. 그중에서  $\phi$ ien의 내적 공간 접근 방법(inner product space approach)과 Lepsy의 반복 적용이 필요없는 복호화 알고리즘이 대표적이다<sup>[11,12]</sup>.

Lepsy의 알고리즘은 복호기에서 반복 적용이 필요없으므로 복호화 시간이 크게 단축된다는 장점이 있으나, 레인지 블럭의 크기가  $8 \times 8$  이하로 제한되므로 화소값이 부드럽게 변하는 영역을 효율적으로 부호화하지 못한다. 본 논문에서는 Lepsy의 알고리즘을 자기 벡터 양자화기(self-VQ)의 관점에서 해석하고, 복호기에서 반복 적용이 필요없을 조건을 일반화하여, 레인지 블럭의 크기에 대한 제한을 완화한다. 또한, 이렇게 일반화된 조건을 이용하여, 넓고 부드럽게 변하는 레인지 블럭을 적은 비트로 효율적으로 부호화하는 새로운 알고리즘을 제안한다. 본 논문의 2 장에서는 프랙탈 영상 압축 기법을 위한 이론적 배경과 기존의 프랙탈 부호화기들을 소개하고, 3 장에서는 복호기에서 반복 적용이 필요없게 하는 조건을 자기 벡터 양자화기의 관점에서 일반화하고, 이를 이용한 새로운 알고리즘을 제안한다. 그리고 4 장에서는 실험 결과 및 검토가, 5 장에서는 결론이 제시된다.

## 2. 프랙탈 부호화 기법

이 장에서는 본 논문의 전개상 필요한 프랙탈 영상 압축 기법의 이론적 배경과, 모든 프랙탈 영상 압축 기법의 원형이라 할 수 있는 Jacquin의 프랙탈 부호화기에 대해서 간략히 설명한다. 그리고 Jacquin의 방법을 개선하기 위하여  $\phi$ ien에 의해 제안된 내적 공간 접근법에 대하여 소개한다.

### 2.1 프랙탈 영상 압축 기법의 수학적 배경

프랙탈 영상 압축 기법의 이론적 배경은 Barnsley에 의해 세워졌다<sup>[9]</sup>. 디지털 영상 공간(digital image space)은 영상들의 집합에 대한 수학적 모델인데, 크기가  $M \times N$ 인 영상들의 집합은  $R^{M \times N}$ 으로 모델링할 수 있다. 이 영상 공간  $I = R^{M \times N}$ 에 속하는 두 원소  $X, Y$  간의 길이를 다음과 같이 정의한다.

$$d(X, Y) = \|X - Y\| = \sum_{m=1}^M \sum_{n=1}^N \{X(m, n) - Y(m, n)\}^2. \quad (1)$$

또한, 영상 공간에서 영상 공간으로의 변환  $f$ 는 다음의 성질을 만족시킬 때 축소 변환이라고 한다.

$$f : I \rightarrow I, \quad 0 \leq s < 1, \\ d(f(X), f(Y)) \leq s \cdot d(X, Y) \text{ for all } X, Y \in I, \quad (2) \\ s : \text{contractivity factor of } f.$$

축소 변환 정리 (contraction mapping theorem)에 의하면<sup>9)</sup>, 축소 변환  $f$ 는 유일한 고정점  $X_f$ 를 가지며, 임의의  $X \in I$ 에 대하여  $f$ 를 반복 적용하면  $X_f$ 로 수렴한다.

$$X_f = f(X_f). \quad (3)$$

$$\lim_{n \rightarrow \infty} f^n(X) = X_f \text{ for all } X \in I. \quad (4)$$

이 축소 변환 정리가 프랙탈 영상 압축 기법의 기반이 되는 이론이다. 즉 부호화하고자 하는 영상  $X$ 를 고정점으로 가지는 축소 변환  $f$ 를 찾으면 복호기에서는  $f$ 의 정보를 받아 임의의 초기 영상에 대하여  $f$ 를 반복 적용하면 원 영상  $X$ 를 얻을 수 있다. 부호화하고자 하는 영상을 고정점으로 가지는 축소 변환 함수를 찾는 것이 역변환 문제 (inverse problem)이다. 원 영상  $X$ 보다  $f$ 가 간략히 표시되면 부호화 이득을 얻는데 원 영상  $X$ 를 정확히 고정점으로 가지는  $f$ 를 찾는 것은 실제적으로 거의 불가능한 일이다. 따라서 역변환 문제를 보다 완화하기 위하여 다음의 콜라주 정리를 이용한다<sup>9)</sup>.

$$d(X_f, X) \leq \frac{1}{1-s} d(X, f(X)), \quad (5)$$

$X_f$  : fixed point of  $f$ .

즉, 원 영상  $X$ 에 대하여 축소 변환  $f$ 를 찾아  $X$ 와  $f(X)$  사이의 거리가  $\epsilon$  보다 작으면,  $X$ 와  $f$ 의 고정점  $X_f$ 사이의 거리는  $\frac{1}{1-s}\epsilon$ 보다 작게 된다. 이는 부호기 측에서 원 영상에 어떤 축소 변환을 적용했을 때 그 결과와 원 영상간의 오차가 크지 않으면, 복호기에서 이 변환을 반복 적용하여 얻은 고정점과 원 영상과의 오차도 위의 관계로 상한이 주어짐을 의미한다.

### 2.2 Jacquin의 프랙탈 영상 부호화기

역변환 문제를 처음으로 해결한 사람이 Jacquin이다

<sup>10)</sup> Jacquin의 알고리즘은 영상을 블럭으로 분할하여 각 블럭에 해당하는 변환을 찾는다. 원 영상  $S$ 가 있을 때 이를 겹치지 않는  $N$ 개의 정사각형 블럭으로 나눈다. 이를 레인지 블럭 (range block)  $\{R_i : 0 \leq i < N\}$ 이라 한다. 각 레인지 블럭을 부호화 하기 위해서, 레인지 블럭의 크기가  $B \times B$ 라면, 원 영상 속에서 크기가  $2B \times 2B$ 인 블럭들 중 레인지 블럭과 가장 유사한 블럭을 찾아 레인지 블럭으로의 변환을 구한다. 이렇게 레인지 블럭을 부호화하기 위하여 찾는 블럭을 도메인 블럭 (domain block)  $\{D_i : 0 \leq i < M\}$ 이라 하는데, 도메인 블럭들은 영상 안에서 서로 겹칠 수 있다.

도메인 블럭으로부터 레인지 블럭으로의 근사는 다음 식과 같다.

$$R_i \simeq \tilde{R}_i = s_i \mathcal{F}_i \circ \mathcal{G}(D_k) + o_i. \quad (6)$$

우선  $G$ (space contraction operator)는 도메인 블럭의 크기가 레인지 블럭의 크기보다 가로, 세로 방향으로 각각 두배씩 크므로 4개의 화소값을 평균하여 1개의 화소값으로 대응시켜 두 블럭의 크기가 서로 같도록 만드는 것이다. 그리고  $F$ (isometry operator)는 이렇게 크기가 줄어든 블럭의 픽셀 값들의 위치를 서로 교환하는 것이다. Jacquin이 사용한 이소메트리 변환의 종류에는 90°씩 회전시키는 네개의 변환과 이를 거울에 반사시킨 네개의 변환, 모두 합해서 8가지가 있다.  $s_i$ 와  $o_i$ 는 각각 이소메트리 변환까지 거친 도메인 블럭의 화소 값들에 곱하고 더하는 상수이다. 이러한 근사 과정이 그림 1에 나타나 있다.

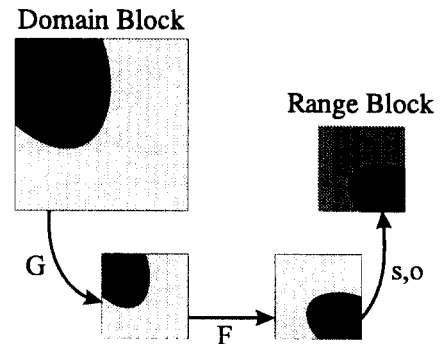


그림 1. 도메인 블럭으로부터 레인지 블럭으로의 변환  
Fig. 1. The domain-range mapping

즉, 레인지 블럭  $R_i$ 를 부호화하기 위해서 모든 도메인 블럭과 이소메트리 변환,  $s, o$  계수에 대하여 위의 변환을 거친 후  $R_i$ 를 가장 잘 근사하는 변환을 찾는다. 이렇게 찾은 변환을  $t_i$ 라 할때 각각의 레인지 블럭  $R_i$ 에 대한  $t_i$ 들의 합이 영상에 대한 변환이 된다.

$$T: I \rightarrow I, \quad T = \sum_{i=0}^{N-1} \tau_i. \quad (7)$$

이 변환  $T$ 는 축소 변환이 되기 위해서 각  $t_i$ 에 해당하는  $s_i$ 의 절대값이 모두 1 보다 작아야 한다. 이렇게 부호화된 정보 (각 레인지 블럭에 대하여 그에 해당하는 도메인 블럭의 위치, 이소메트리 변환,  $s, o$  계수)를 가지고, 복호기에서는 임의의 초기 영상에 대하여 변환을 적용하여 새로운 영상을 만들고 이 새로운 영상에 다시 같은 변환을 적용하는 것을 여러번 반복하면 원 영상의 근사 영상을 얻는다.

Jacquin의 프랙탈 부호화기의 단점은 영상을 부호화하기 위하여 각 레인지 블럭마다 매우 많은 경우의 변환 (도메인 블럭의 위치, 이소메트리 변환과  $s, o$  계수값들의 조합으로 이루어진 모든 변환)을 수행하여 최적의 변환을 찾기 때문에 많은 시간이 걸린다는 점이다. Jacquin은 이를 해결하기 위하여 도메인 블럭과 레인지 블럭을 Shade, Midrange 그리고 Edge 로 분류하는 방법을 택하여 한 레인지 블럭을 부호화하기 위하여 같은 영역에 속하는 도메인 블럭들 안에서만 최적의 변환을 찾았다. Shade 블럭은 블럭안의 화소값이 거의 변화하지 않는 블럭이고, 화소값이 부드럽게 변화하는 블럭을 Midrange 블럭, 급격히 변화하는 블럭을 Edge 블럭으로 분류하였다. 이런 분류를 통하여, 부호화 시간이 감소하고, 영상의 예지가 잘 보존되어 좋은 주관적인 화질을 보인다. 그러나 부호화 시간이 다른 부호화 기법에 비하여 여전히 매우 길다는 단점이 있다.

2.3 내적 공간 접근법(Inner product space approach)

Oien 등은 내적 공간 접근법을 사용하여 식 (6)의  $s, o$  계수를 정확하게 구하는 방법을 제안하였다<sup>[11]</sup>. 내적 공간 접근법은 식 (6)에서 레인지 블럭의 근사가 이소메트리 변환까지 거친 도메인 블럭  $F, oG(D_k)$ 과 모든 화소값이 1 인 상수 블럭과의 일차 결합 (linear combination)으로 이루어진다는 성질을 이용한다. 즉, 이소메트리까지 거친 도메인 블럭을  $D$ 이라 하고 상수 블럭

을  $C$ 라 하면 레인지 블럭의 근사 ( $\tilde{R}$ )는 다음과 같이 나타낼수 있다.

$$\tilde{R} = s \cdot D + o \cdot C. \quad (8)$$

최적의  $s, o$  계수를 찾는 것은  $D$ 와  $C$ 를 기저로 하는 부분 공간상의 원소들 중에서  $R$ 과 가장 가까운 거리에 있는 원소  $\tilde{R}$ 을 찾는 것이다.  $D$ 와  $C$ 는 보통의 경우 직교 (orthogonal) 하지 않는 경우가 대부분인데 이를 Gram-Schmidt 직교화 과정[18]을 통하여 서로 수직이 되도록 만든다. 이런 직교화 과정은  $D$ 블럭의 모든 화소값들의 평균을 각각의 화소에서 빼서 새로운 기저  $\hat{D}$ 로 만드는 것이다. 이 과정을 통하여  $\hat{D}$ 의 화소값들의 평균은 0이 되고,  $\hat{D}$ 는  $C$ 에 직교하게 된다.

$$\hat{D} = D - \frac{\langle D, C \rangle}{\langle C, C \rangle} C. \quad (9)$$

$$\hat{R} = \hat{s} \cdot \hat{D} + \hat{o} \cdot C. \quad (10)$$

여기서  $\langle A, B \rangle$ 는  $A$ 와  $B$ 의 내적을 나타낸다. 그러면, 선형 대수의 정사영 정리 (orthogonal projection theorem)에 따라  $\hat{R}$ 과  $R$ 사이의 거리를 최소화하는 최적의  $\hat{s}, \hat{o}$  계수는 레인지 블럭  $R$ 을 각각  $\hat{D}$ 와  $C$ 에 정사영시킴으로써 얻을 수 있다.

$$\hat{s} = \frac{\langle \hat{D}, R \rangle}{\langle \hat{D}, \hat{D} \rangle}, \quad \hat{o} = \frac{\langle C, R \rangle}{\langle C, C \rangle}. \quad (11)$$

이런 내적 공간 접근법을 이용하면 입력 영상이나 최종적으로 구해진 변환식의 계수값에 상관없이, 복호기측에서 일정한 횟수만큼 변환을 반복적용하면 고정점으로 정확히 수렴함이 알려져 있다<sup>[13]</sup>. 레인지 블럭의 크기가  $2^h \times 2^h$ 이고 그에 해당하는 도메인 블럭의 크기가  $2^d \times 2^d$ 이면 반복 적용의 횟수  $K$ 는 다음의 식으로 정해진다.

$$K = \max_i \left\lceil \frac{b_i}{d_i - b_i} \right\rceil. \quad (12)$$

여기서  $\lceil a \rceil$ 는  $a$ 보다 작지 않은 최소의 정수를 의미한다.

Lepson y 등은 위의 식 (12)를 이용하여 복호기에서 반복 적용이 필요없는 프랙탈 부호화 기법을 제안하였다<sup>[12]</sup>. 이들은 레인지 블럭의 크기를  $4 \times 4, 8 \times 8$ 로 한정하고, 이에 해당하는 도메인 블럭의 크기를 각각  $16 \times 16$ .

64×64로 하여 위의 식에서  $K = 1$ 의 조건을 만족시킨다. 그러나, 복호기에서 반복 적용이 필요없기 위해서는 도메인 블록의 크기가 가로, 세로 방향으로 각각 레인지 블록의 크기의 제곱이어야 하므로 Lepa \* y의 알고리즘은 레인지 블록의 크기가 제한된다. 만약 16×16크기의 레인지 블록을 사용하고자 하면 도메인 블록의 크기가 256×256이어야 하므로 실제 적용이 거의 불가능하다. 왜냐하면, 16×16블럭과 256×256블럭간에 유사성이 있는 경우가 아주 드물기 때문이다. 그러나 복호기에서 반복 적용이 필요없다는 성질은 매우 중요한 장점이다. 복호화 시간이 매우 빠르기 때문에 영상 CD-ROM 과 같이 부호화는 한 번 이루어지지만 복호화는 여러번 이루어지는 응용에 사용될 수 있다.

### 3. 제안하는 프랙탈 영상 압축 기법

본 논문에서는 복호기에서 반복적용이 필요없는 새로운 프랙탈 부호화기를 제안한다. Lepa \* y의 알고리즘은 8×8보다 큰 크기의 레인지 블록을 사용하지 못하므로, 화소값이 부드럽게 변화하는 영역을 효율적으로 부호화하지 못한다는 단점이 있었다. 그러나 본 논문에서는 자기 벡터 양자화기의 관점에서 복호기에서 반복 적용이 필요없게 하는 조건을 일반화하고, 이 조건을 이용하여 새로운 알고리즘을 제안한다.

#### 3.1 자기 벡터 양자화기 관점에서의 프랙탈 부호화 기법

Lepa \* y의 부호화 알고리즘에서, 복호기측 반복 적용이 필요없는 까닭은 다음과 같은 관점에서 볼 수 있다. 레인지 블록의 크기가 모두 4×4라고 가정하자. 그러면 도메인 블록의 크기는 16×16이고, 이는 4×4개의 레인지 블록으로 이루어져 있다. 레인지 블록을 부호화하기 위하여 우선 식 (10)의  $\delta$  계수를 구한다. 이  $\delta$  계수는 레인지 블록의 화소값의 평균 즉, DC 값에 해당한다. 나머지 AC에 해당하는 정보는 이 레인지 블록에 대응되는 도메인 블록의 위치와 그에 곱해지는  $\delta$  계수에 의해서 전달된다. 부호기 측에서 도메인 블록으로 레인지 블록을 근사시키기 위해서는 4×4개의 화소를 평균하여 1 개의 화소로 만들어(공간 축소 변환), 두 블록의 크기를 같도록 하는데, 복호기측에서는 그 평균값이 이미 그 4×4개의 화소에 해당하는 레인지 블록의 계수  $\delta$ 에 의해 알려져 있다. 즉 공간 축소 변환된 도메인 블록의 화소값들이 그 도메인 블록을 이루는 16 개의 레인지 블록의  $\delta$  계수들의 형태로 부호기에서 복호기로 전달된다. 따라서 복호기측에서 단 한번의 프랙탈 변환만으로 영상을 반복 적용없이 복호화할 수 있다. 그림 2에 이 과정이 나타나 있다. 그림 2 (a)에 레인지 블록 (진한 선으로 둘러 싸인 작은 사각형)과 이 레인지 블록을 부호화하기 위한 도메인 블록이 나타나 있다. 그림 2 (a)는 복호기에서  $\delta$  계수, 즉 레인지 블록의 DC 값만으로 복원된 결과이다. 그림 2 (b)는 이 도메인 블록을 공간 축소 변환한 블록이고, 그림 2 (c)는 이를 이용하여 최종적으로 복원된 레인지 블록이 아직 복원되지 않은 다

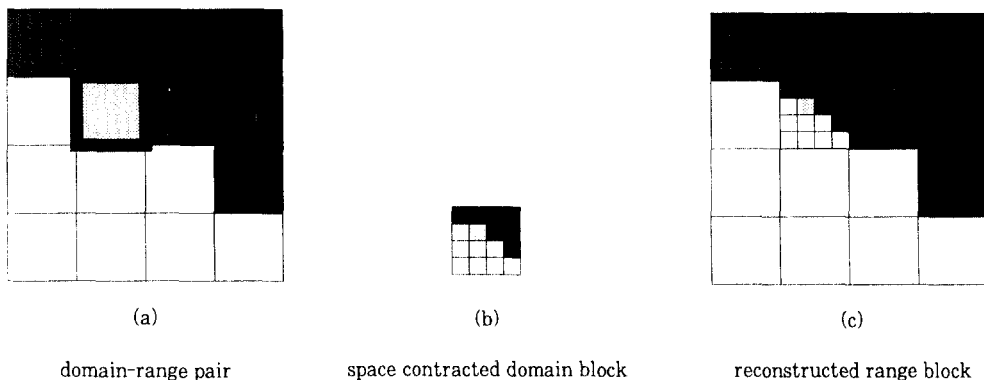


그림 2. Lepa \* y 알고리즘의 도식 설명  
Fig. 2. Illustration of Lepa \* y's algorithm

른 레인지 블록들과 함께 나타나 있다.

복호기에서 반복 적용이 필요 없는 프랙탈 부호화 기법은 이미 프랙탈의 여러 성질을 잃어버린다. 그 중 제일 중요한 점은 해상도에 상관없는 이미지를 복원할 수 없다는 것인데, 위의 알고리즘으로 구한 변환을 복호기에서 가로, 세로 각각 2배의 크기의 영상에 적용하면 4개의 픽셀이 상수 블록으로 됨을 알 수 있다. 즉, 같은 크기로 복호화된 영상을 0-차 인터폴레이션(zeroth order interpolation)한 결과를 얻는다. 따라서 복호기에서 반복 적용이 필요 없는 알고리즘은 프랙탈의 성질보다는 코드북을 부호화하려는 영상에서 발생시킨다는 의미에서 자기 벡터 양자화기 (self-VQ)<sup>[14]</sup>의 특성을 가진다. VQ의 관점에서 위의 알고리즘을 살펴보면 영상을 부호화하기 위하여 레인지 블록으로 분할한 후에, 각 레인지 블록의 평균값을 구하고 이 평균값들로 이루어진 영상에서 가상적인 코드북(virtual codebook)<sup>[10]</sup>을 만들어 레인지 블록의 AC 정보를 부호화하는 것이다. 즉 도메인 블록의 위치는 이 코드북으로의 인덱스(index)를 의미하는 것이다. 이런 프랙탈 부호화 기법의 잠재적인 가능성은 이미지의 자기 유사성(self-similarity)을 이용하여 영상에 적응적인 코드북을 만들고, 그 코드북은 이미 DC 정보에 내재되어 있으므로 부호화 효율을 높일 수 있다는 특성에 있다.

본 논문에서는 이런 개념을 확장하여 다음과 같은 새로운 프랙탈 영상 압축 기법을 제안한다. 우선, 영상에 레인지 블록별로 세계의 고정 블록(fixed basis block)을 사용하여 근사한 후, 근사된 영상으로부터 특성이 다른 두개의 코드북을 발생시킨다. 첫째 코드북은 화소값이 부드럽게 변화하는 레인지 블록을 부호화하기 위하여 스무딩 변환을 이용하여 발생시키고, 둘째 코드북은 예지같이 화소값이 급격히 변화하는 레인지 블록을 부호화하기 위하여 공간 축소 변환을 이용하여 발생시킨다. 이와 같이 특성이 다른 두개의 코드북을 사용함으로써, 레인지 블록의 크기에 대한 제한 조건이 완화되고, 보다 효율적인 영상 부호화를 가능하게 한다. 제안하는 부호화 알고리즘은 다음 절에서 상세히 설명한다.

### 3.2 부호기 알고리즘(encoding algorithm)

제안하는 알고리즘은 부호화 효율을 높이기 위하여 가변 블록 크기를 사용한다. 즉 영상의 복잡도에 따라 블록의 크기를 변화시켜, 화소값이 급격히 변하는 영역은

작은 블록으로, 평탄한 영역은 큰 블록으로 분할하여, 특성에 맞는 부호화 방법을 사용한다. 기존의 프랙탈 영상 압축 기법은 가변 블록 크기를 사용하기 위하여, 각 레인지 블록에 해당하는 프랙탈 변환을 찾아 그 근사 오차가 일정한 문턱값을 넘으면 작은 레인지 블록들로 나누는 방법을 택하여 부호화 시간이 길어지는 단점이 있었다. 이에 비하여 제안하는 알고리즘은 전처리 과정에서 프랙탈 근사보다 훨씬 빠른 고정 블록 근사를 이용하여 영상을 가변 블록 크기로 분할한 후에 각각의 레인지 블록을 부호화하므로 부호화 시간이 단축된다.

제안하는 부호화 과정은 다음의 세단계로 나뉜다.

- 전처리 과정(preprocessing)
- 16×16, 32×32 블록 부호화
- 4×4, 8×8 블록 부호화

전처리 과정 : 전처리과정에서는 영상을 레인지 블록으로 분할한다. 영상의 복잡도에 따라 최대 32×32 크기의 레인지 블록에서 최소 4×4 크기의 레인지 블록으로 분할된다. 영역 분할을 위하여 부가 정보량이 작은 Quadtree Decompsition<sup>[15]</sup>을 사용하는데 우선 32×32블록을 평면으로 근사(planar approximation)하여 그 근사 오차가 일정한 문턱값을 넘으면 4개의 16×16블록으로 나누고 이들을 다시 평면 근사하여, 같은 과정을 4×4 크기의 블록이 생성될 때까지 반복한다. 평면 근사는 X, Y, C3 개의 고정 블록을 사용하여 레인지 블록을 근사하는 과정이다. X, Y블록은 각각 화소값이 x, y 방향으로 증가하는 블록이고 C블록은 상수블록인데 레인지 블록의 크기가 r×r이라면 다음과 같이 정의하여 서로 직교하게 만든다.

$$\begin{aligned} X(m, n) &= 2m-r+1, \\ Y(m, n) &= 2n-r+1, \quad (0 \leq m, n < r), \\ C(m, n) &= 1, \end{aligned} \quad (13)$$

이렇게 서로 직교하는 고정 블록에 레인지 블록 R을 정사영하면 최적의 평면 근사  $R_{fixed}$ 를 얻는다.

$$\begin{aligned} R_{fixed} &= \alpha \cdot X + \beta \cdot Y + \gamma \cdot C \\ &= \frac{\langle X, R \rangle}{\langle X, X \rangle} \cdot X + \frac{\langle Y, R \rangle}{\langle Y, Y \rangle} \cdot Y \\ &\quad + \frac{\langle C, R \rangle}{\langle C, C \rangle} \cdot C. \end{aligned} \quad (14)$$

이 평면 근사를 통하여 화소값이 부드럽게 변하는 영역은 큰 레인지 블록으로, 예지같이 화소값이 급격히 변

화하는 영역은 작은 레인지 블럭으로 분할된다. 또한 이렇게 평면 근사된 영상은 공간 축소 변환과 스무딩 변환을 거쳐 각각의 레인지 블럭을 부호화하기 위한 코드북으로 사용된다. 공간 축소 변환된 코드북은 화소값이 급격히 변화하는 4×4 또는 8×8크기의 레인지 블럭을 부호화하기 위하여 사용되고, 스무딩 변환된 코드북은 화소값이 부드럽게 변하는 16×16 또는 32×32 크기의 레인지 블럭을 부호화하기 위하여 사용된다.

16×16, 32×32 블럭 부호화 : 전처리 과정에서 평면 근사를 통하여 16×16, 32×32 크기로 분할된 레인지 블럭은 화소값이 부드럽게 변하는 블럭이다. 그러나 평면 근사된 영상은 블럭간의 경계에서 구획 현상(blocking effect)이 나타나 주관적인 화질을 손상한다. 이러한 단점을 극복하기 위하여 평면 근사된 영상( $I_{fixed}$ )을 저대역 제한 필터를 통과시켜 얻은 영상( $I_{low}$ )으로부터 도메인 블럭을 찾아 주어진 레인지 블럭을 부호화한다. 즉, 전처리 과정에서 나온 영상은 각각의 블럭이 평면으로 이루어져 있으나, 이를 스무딩 변환(저대역 제한 필터)을 적용하여 곡면으로 만든 후에 이를 새로운 코드북으로 삼아, 화소값이 부드럽게 변하는 16×16, 32×32 블럭을 부호화한다. 사용하는 저대역 제한 필터는 9×9 moving average 필터이다. 도메인 블럭의 크기는 레인지 블럭의 크기와 같게 할 때 레인지 블럭을 가장 잘 근사함을 실험적으로 알 수 있었다. 이 과정을 1 차원 신호에 대하여 나타낸 것이 그림 3이다. 그림 3 (a)는 부호화 하고자 하는 신호이고, 그림 3 (b)는 이 신호를 2 차원의 평면 근사에 해당하는 선형 근사한 결과이다. 그림 3 (c)는 이 선형 근사된 신호를

저대역 제한 필터를 통과시킨 신호이고, 이를 이용하여 그림 3 (d)에서처럼 주어진 신호를 부호화한다.

레인지 블럭  $R$ 의 근사는  $I_{low}$ 로부터 찾은 최적의 도메인 블럭을  $D$ 라 할 때  $D$ 와 식 (14)의  $C$ 블럭으로 근사하는데, 우선  $D$ 와  $C$ 를 직교화 한 후에 다음과 같이 구한다.

$$\widehat{D} = D - \frac{\langle D, C \rangle}{\langle C, C \rangle} \cdot C. \tag{15}$$

$$\begin{aligned} R &= s \cdot \widehat{D} + \gamma \cdot C \\ &= \frac{\langle \widehat{D}, R \rangle}{\langle \widehat{D}, \widehat{D} \rangle} \cdot \widehat{D} + \gamma \cdot C. \end{aligned} \tag{16}$$

전처리 과정에서 나온 식 (14)의  $\alpha$ ,  $\beta$ ,  $\gamma$ 계수는 모두 복호기로 전달되는데, 위와 같이 최종적으로 레인지 블럭을 근사하기 위하여 상수블럭에 해당하는  $\gamma$ 계수만을 사용하는 까닭은 도메인 블럭  $D$ 를  $X, Y, C$ 블럭에 모두 직교화 하도록 하는 과정이 시간이 너무 많이 걸리다는 단점이 있고, 계산이 너무 복잡하여 그 연산 오차가 누적되어 실제 화질 향상에 도움이 되지 않기 때문이다.

4×4, 8×8블럭 부호화 : 전처리 과정에서 4×4 또는 8×8로 분할된 레인지 블럭은 블럭 내부의 화소값이 급격히 변화하여 평면 근사의 오차가 큰 블럭들이다. 주로 에지로 이루어진 영역이 많은데 이런 블럭을 부호화하기 위해서는 전처리 과정에서 나온  $I_{fixed}$  영상에서 크기가 레인지 블럭 크기의 제곱인 도메인 블럭을 찾아 부호화한다. 즉 전처리 과정에서 나온  $I_{fixed}$  영상을 공간 축소 변환하여 코드북으로 이용한다. 식 (15), (16)과 마찬가지로 공간 축소 변환된 도메인 블럭과 상수 블럭

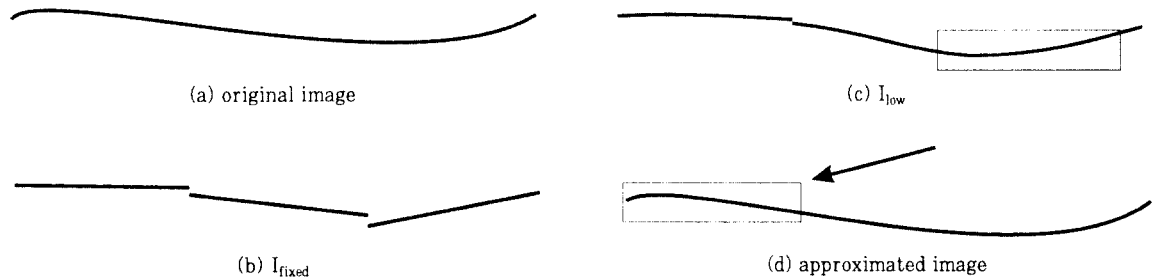


그림 3. 부드럽게 변하는 신호의 부호화(1차원의 경우)  
Fig. 3. Encoding of smooth signal(in case of 1-D signal)

으로 이루어진 공간에서 레인지 블록을 최적으로 근사하는 변환을 찾는다. 이렇게 도메인 블록과 상수 블록으로 근사화된 결과와 전처리 과정에서  $X, Y, C$  블록으로 근사화된 결과를 비교하여 PSNR의 관점에서 더 좋은 근사를 부호화에 사용한다.

### 3.3 복호기 알고리즘 (decoding algorithm)

복호기의 알고리즘은 부호기의 알고리즘처럼 도메인 블록을 찾는 과정이 없어 간단하다. 우선, Quadtree 정보와  $\alpha, \beta, \gamma$  계수를 가지고, 전처리된 영상  $I_{fixed}$ 을 복원하고, 이를 부호기와 같은  $9 \times 9$  moving average lowpass filter 를 사용하여  $I_{low}$ 를 만든다.  $16 \times 16, 32 \times 32$  블록은  $I_{low}$ 로부터 부호기에서 보낸 변환 정보 (도메인 블록의 위치와  $s$  계수)를 적용하여 복원하고,  $4 \times 4, 8 \times 8$  블록은 마찬가지로  $I_{fixed}$ 로부터 복원한다.

### 3.4 계수 양자화와 비트 할당

부호화 과정에서 찾은 최적의 계수들은 양자화되어 있

지 않아 전송이나 저장을 하기위하여서는 이 계수들의 양자화와 효율적인 비트 할당이 필요하다.

우선, 전처리 과정에서는 레인지 블록별  $\alpha, \beta, \gamma$  계수를 양자화하여야 한다. 그런데  $\gamma$  계수는 레인지 블록의 평균 화소값을 나타내는 것이므로 0~255의 레벨을 나타내기 위하여 8 비트를 할당한다. 또한,  $X, Y$  블록에 해당하는 계수인  $\alpha, \beta$ 는 3개의 비트를 할당하여 Lloyd-Max 양자화하였다. 그런데 계수들의 분포가 레인지 블록의 크기에 따라 다르므로, 그 크기에 따른 계수들의 분포를 10 개의 테스트 영상으로부터 얻어 각각 양자화하였다. 그림 4에 레인지 블록 크기에 따른  $\alpha, \beta$  계수의 확률 분포가 나타나 있다. 계수들의 분포가 블록 크기에 따라 다른 이유는, 전처리 과정에서 큰 블록으로 분할된 레인지 블록은 그 화소값들이 부드럽게 변화하므로 최적의  $\alpha, \beta$  계수도 주로 0 주변에 모여있지만, 화소값들이 급격히 변화하는 작은 레인지 블록에 해당하는 계수들은 넓게 퍼져 있기 때문이다.

레인지 블록의 실제 코딩에서는 도메인 블록의 위치와

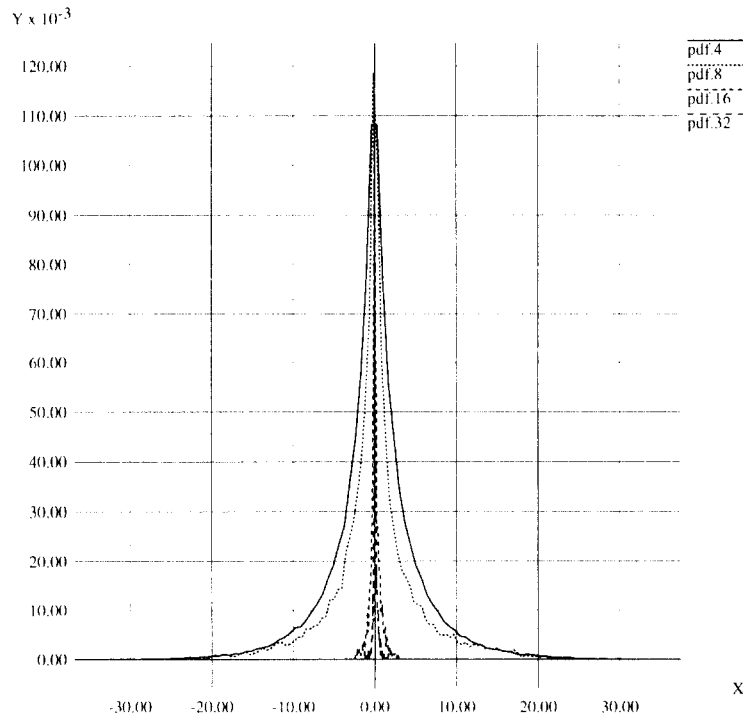


그림 4.  $\alpha, \beta$  계수의 확률 분포

Fig. 4. The probability distribution function of  $\alpha, \beta$  coefficients



그에 해당하는  $s$  계수의 양자화가 필요하다.  $16 \times 16$ ,  $32 \times 32$  크기의 레인지 블록은 저대역 제한 필터를 거친 전처리 결과에서 도메인 블록을 찾는데, 부호화 시간과 비트 할당을 고려하여 레인지 블록 주변의 25 가지 도메인 블록 후보들 중에서 최적의 도메인 블록을 찾는다. 이 25 가지 도메인 블록의 위치는 그림 6에 나타나 있다. 이 25 가지의 경우를 부호화하기 위하여 5 비트를 할당하였다. 그리고,  $s$  계수의 양자화는 10 개의 테스트 영상에서 구한 확률 분포에 5 비트를 할당하여 Lloyd-Max 양자화하였다.  $4 \times 4$ ,  $8 \times 8$  크기의 블록도 마찬가지로 16 개의 도메인 블록 후보들을 사용하여, 도메인 블록의 위치를 부호화하는데 4 비트,  $s$  계수를 양자화하는데 5 비트를 할당하였다. 그림 5에  $32 \times 32$ ,  $16 \times 16$ 과  $8 \times 8$ ,  $4 \times 4$  크기에 해당하는 레인지 블록의  $s$  계수들의 확률 분포가 각각 나타나 있다.  $32 \times 32$ ,  $16 \times 16$  크기의 레인지 블록은 저대역 필터를 거친 영상에서 같은 크기의 도메인 블록을 찾으므로,  $s$  계수가 제곱 크기의 도메인 블록을 찾는  $8 \times 8$ ,  $4 \times 4$  크기의 레인지

블록의 경우에 비하여 1 주변에 몰려 있음을 확인할 수 있다.  $s$  계수를 양자화하면 최적의 계수를 사용하는 것보다 복호화된 영상의 화질이 필연적으로 나빠지지만 5 내지 6 비트를 사용하면 그 화질 저하의 정도가 매우 낮다는 것이 알려져 있다<sup>[16]</sup>. 실제 실험에서도 최적의 계수를 사용했을때보다 약 0.1 dB 정도의 화질 저하만이 관찰되었다.

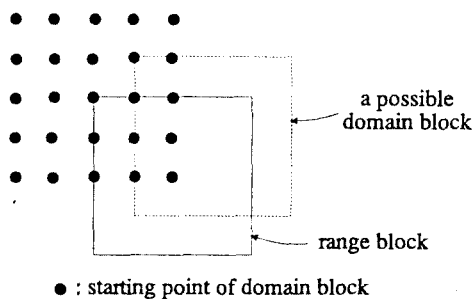


그림 6. 도메인 블록의 위치  
Fig. 6. The location of domain block

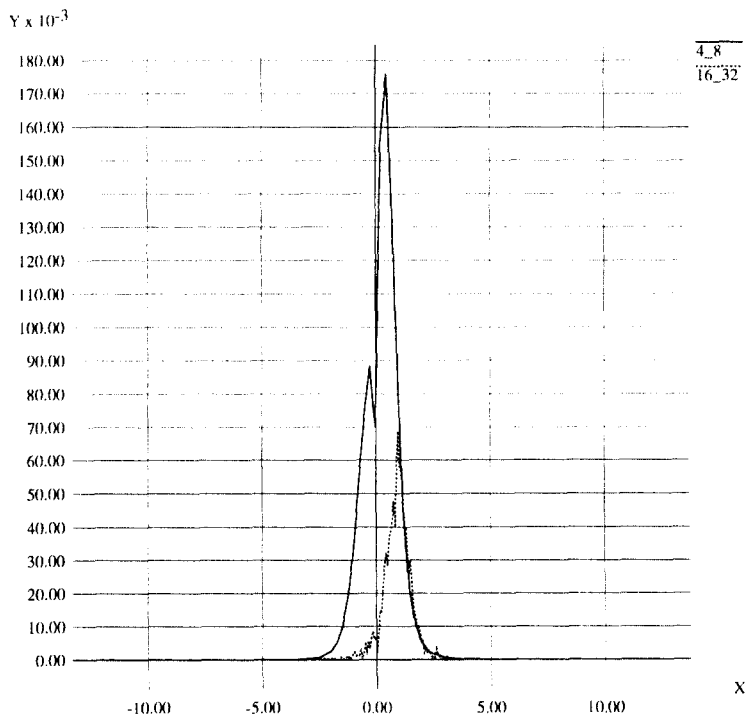


그림 5.  $s$  계수의 확률 분포  
Fig. 5. The probability distribution function of  $s$  coefficients

표 1. 각 계수들에 대한 비트 할당  
Table 1. Bit allocation to each coefficient

전처리 과정	$\alpha$	$\beta$	$\gamma$	total
	3	3	8	14 bit
16×16, 32×32 레인지 블록	도메인 블록의 위치		s	10 bit
	5		5	
4×4, 8×8 레인지 블록	도메인 블록의 위치		s	9 bit
	4		5	

제안하는 알고리즘의 비트 할당이 표 1에 제시되어 있다.

#### 4. 실험 결과 및 검토

제안하는 반복 적용이 필요없는 프랙탈 영상 압축 기법을 컴퓨터 모의 실험을 통하여 기존의 방법과 비교, 평가하였다. 테스트 영상으로써 512×512 크기의 영상들을 사용하였다. 그림 7에 테스트 영상 중 하나인 Lena 원 영상을 나타내었다. 이 Lena 영상을 부호화하기 위하여 우선 평면 근사를 통하여 Quadtree 구조로 분할하는데 그 분할된 결과가 그림 8이다. 또한 전처

리 과정에서 3 개의 블록으로 평면 근사된 결과 ( $I_{fixed}$ )와 이를 저대역 제한 필터를 거친 결과 ( $I_{low}$ )가 나오는데 각각 그림 9 (a)와 9 (b)에 나타나 있다. 이 전처리 과정에서 생성된  $I_{fixed}$ 와  $I_{low}$ 가 각각 8×8 이하와 16×16 이상의 레인지 블록을 부호화하기 위한 코드북으로 사용되는데, 그림에서도 볼 수 있는 바와 같이  $I_{low}$ 는 그 화소값이 부드럽게 변화하므로 넓고 화소값이 부드럽게 변하는 레인지 블록을 효율적으로 부호화한다. 이렇게 부호화된 결과를 그림 10 (a)에 나타내었다. 비교 대상으로 Lepsoy의 알고리즘을 사용하여 제안하는 알고리즘과 동일한 조건하에서 같은 전송율을 가질 때의 결과를 그림 10 (b)에 나타내었다. 제안하는 알고리즘은 Lepsoy의 알고리즘보다 넓은 레인지 블록을 사용함에도 불구하고, 같은 전송율에서 구획 현상 (blocking effect) 없이 영상을 복원하여, 주관적인 화질이 좋을 수 있다. 그림 11 (a)와 11 (b)에 각각 Zelda 원 영상과 제안하는 알고리즘으로 부호화된 결과를 나타내었다.

여러 가지 512×512 테스트 영상에 적용한 결과를 표 2에 나타내었다. Lepsoy 알고리즘의 결과는 [12]에서 부호기의 구현 방법에 대한 언급이 없이 결과만 제시되어 있으므로 그 수치를 그대로 인용하고, 제안하는 알고리즘과 같이 s 계수에 5 비트, o 계수에 8 비트를 할



그림 7. 모의 실험에 사용된 Lena 영상  
Fig. 7. The original Lena image

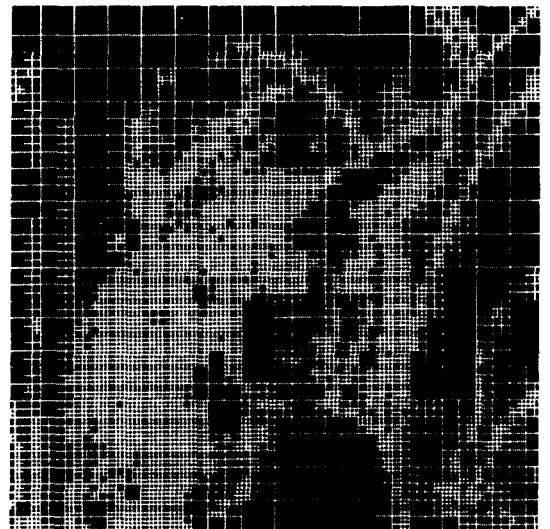


그림 8. Quadtree 구조로 분할된 Lena 영상  
Fig. 8. The quadtree decomposed Lena image



그림 9. Lena 영상의 전처리 결과  
Fig. 9. The preprocessed results of Lena image



그림 10. 결과 영상의 비교  
Fig. 10. The comparison of decoded image

당하여 양자화하고, 도메인 블록의 탐색 영역을 레인지 블록 주변으로 제한했을 때의 실험 결과를 괄호안에 제시하였다. 두가지 경우 모두  $Lep * y$ 의 알고리즘에 비

하여 제안하는 알고리즘이 Lena 영상과 Pepper 영상에서 전송률 대 화질 (bit-rate vs. PSNR)의 관점에서 많은 향상이 있음을 알 수 있다. 이는 Lena 영상이



(a) 원영상



(b) 부호화된 결과, 0.29 bpp, 34.1 dB

그림 11. Zelda 영상에 대한 모의 실험 결과  
 Fig. 11. The simulation result on the Zelda image

표 2. Lepp & y 알고리즘과 제안하는 알고리즘의 결과 비교  
 Table 2. Comparison of results between Lepp & y' algorithm and the proposed algorithm

	Lepp & y 알고리즘		제안하는 알고리즘	
	PSNR(dB)	bit rate(bpp)	PSNR(dB)	bit rate(bpp)
Lena	32.0 (29.6)	0.53 (0.41)	32.0	0.41
Pepper	31.5 (29.4)	0.53 (0.39)	32.2	0.38
Boats	31.2 (28.8)	0.59 (0.51)	31.1	0.54

나 Pepper 영상의 화소값들이 상대적으로 부드럽게 변하기 때문이다. Boats 영상과 같이 복잡한 영상에 제안하는 알고리즘을 사용하여도 약간의 향상이 있음을 알 수 있다.

프랙탈 부호화 기법은 영상에 내재하는 자기 유사성을 이용하여 영상을 부호화하는데, 복호기에서 반복 적용이 필요한 알고리즘과 필요없는 알고리즘 모두 각 레인지 블록을 부호화하기 위하여 레인지 블록과 가장 유사한 도메인 블록의 위치에 해당하는 정보가 필요하다. 제안하는 알고리즘에서  $32 \times 32$  레인지 블록이나  $16 \times 16$  레인지 블록을 부호화하기 위해서 그림 6에서처럼 도메인 블록을 그 레인지 블록의 위치 주변에서만 찾는다. 그 까닭은 제안하는 알고리즘의 특성상  $32 \times 32$ ,  $16 \times 16$

레인지 블록을 부호화하기 위하여 같은 크기의 도메인 블록을 사용하므로, 코드북  $I_{fixed}$  상에서 최적의 도메인 블록이 레인지 블록 주변에 있을 확률이 매우 크기 때문이다. 따라서 도메인 블록의 크기가 레인지 블록의 크기보다 2 배 이상인 다른 프랙탈 부호화 기법보다 도메인 블록을 찾는 영역이 줄어들기 때문에, 부호화 시간이 단축되고, 그 위치를 부호화하기 위한 비트 량의 감소가 가능하다.

또한, 도메인 블록과 레인지 블록간의 변환에 직교 변환을 사용하지 않는 대다수의 프랙탈 부호화기가 복호기에서 5~6 회의 반복 적용이 필요한 것에 반하여, 제안하는 알고리즘은 반복 적용이 필요없으므로 그 구현 및 복호화 속도의 관점에서 장점을 가지고 있다. Lepp & y

의 알고리즘은 압축된 영상을 복호화하기 위하여 화소당 덧셈 연산 3 회와 곱셈 연산 1 회가 필요하다<sup>(12)</sup>. 제안하는 알고리즘은 이 연산 이외에 3 개의 고정 블럭을 사용함으로써 부가되는 화소당 2 회의 덧셈연산이 더 필요하고, 저대역 제한 필터링을 하는 연산 과정이 필요하다. 따라서, Lepsoy의 알고리즘에 비하여 약간의 부가 연산이 필요하지만 다른 대다수의 반복적용이 필요한 프랙탈 부호화기가 복호기에서 2 회의 반복 적용을 하는 시간안에 복호화가 가능하다.

### 5. 결 론

본 논문에서는 기존의 반복 적용이 필요없는 프랙탈 부호화기를 자기 백터 양자화기의 관점에서 보고, 그 개념을 확장하여 새로운 프랙탈 부호화기를 제안하였다. 제안하는 알고리즘은 부호화하려는 영상을 평면 근사한 후에, 이로부터 두개의 코드북을 발생시켜, 레인지 블럭의 특성에 따라 알맞은 코드북을 선택하는 방법이다. 즉, 화소값이 부드럽게 변화하는 레인지 블럭은 평면 근사된 영상을 저대역 제한 필터를 거친 영상으로부터 코드북을 발생시켜 부호화하였고, 예지같이 그 화소값이 급격히 변화하는 레인지 블럭은 평면 근사된 영상을 공간 축소 변환한 영상으로부터 코드북을 발생시켜 부호화한다. 이렇게 두개의 코드북을 영상 부호화에 사용함으로써, 레인지 블럭의 크기에 대한 제한 조건이 완화되고, 특히 화소값이 부드럽게 변하는 영역을 효율적으로 부호화한다. 제안하는 알고리즘은 대다수의 기존의 프랙탈 부호화기<sup>(10, 12, 17)</sup> 보다 객관적인 화질(objective quality : PSNR)이나 주관적인 화질에서 향상된 성능을 보였다. 또한 제안하는 프랙탈 부호화기는 영상을 부호화하기 위하여 꼭 축소 변환의 조건을 만족시킬 필요가 없고, 반복 적용이 필요없기 때문에 복호화 시간이 크게 단축된다는 장점이 있다.

### 참고문헌

1. R. Clarke, *Transform Coding of Images*, Academic Press, New York, USA, 1985.
2. A. Gersho and R. M. Gray, *Vector Quantization and Signal Processing Compression*, Kluwer Academic Publishers,

- Norwell, MA, USA, 1992.
3. Joint Photographic Expert Group, *JPEG technical specification, Revision 8*, Technical report, ISO/IEC/JTC1/SC1/WG8, August, 1990.
4. Motion Picture Expert Group, *Recommendation H.261 : Video codec for audiovisual services at p×64 kbits/s*, Technical report, CCITT, 1990.
5. Motion Picture Expert Group, *MPEG-I Draft International Standard 11172*, Technical report, ISO/IEC, March, 1992.
6. Motion Picture Expert Group, *MPEG Draft International Standard*, Technical report, CCITT SGXV, March, 1994.
7. I. Corset, S. Jeannin and L. Bouchard, "MPEG-4: Very low bit rate coding for multimedia applications," in *Proc. of SPIE-VCIP*, vol 2308, pp.1065-1073, 1994.
8. H. Peitgen, H. Jurgens and D. Saupe, "Encoding Images By Simple Transformations," in *Chaos and Fractals: New Frontiers of Science*, pages 229-296, Springer-Verlag, New York, 1992.
9. M. F. Barnsley, *Fractals Everywhere*, Academic Press, San Diego, 1988.
10. A. E. Jacquin, "Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations," *IEEE Trans. on Image Processing*, vol. 1, no. 1, pp.18-30, January 1992.
11. G. E. øien, S. Lepsoy and T. A. Ramstad, "An Inner Product Space Approach to Image Coding by Contractive Transformations," in *Proc of ICASSP*, pp.2773-2776, 1991.
12. S. Lepsoy, G. E. øien and T. A. Ramstad, "Attractor Image Compression with a Fast Non-iterative Algorithm," in *Proc. of ICASSP*, vol. 5, pp.337-340, 1993.
13. G. E. øien and S. Lepsoy, "A Class of Fractal Image Coders with Fast Decoder Convergence," in *Fractal Image Compression-Theory and Application*, chap 8, pp.137-152,

Springer-Verlag, New York, 1995.

14. Y. Fisher, T. P. Shen and D. Rogovin, "Fractal(Self-VQ) Encoding of Video Sequences," in *Proc. of SPIE-VCIP*, vol 2308-134, September, 1994.
15. E. Shusterman and M. Feder, "Image Compression via Improved Quadtree Decomposition Algorithms," *IEEE Trans. on Image Processing*, vol. 3, no. 2, pp.207-215, March, 1994.
16. G. E. Øien, "Parameter Quantization in Fractal Image Coding," in *Proc. of ICIP*, vol. 3, pp.142-146, 1994.
17. Y. Fisher (editor), *Fractal Image Compression-Theory and Application*, Springer-Verlag, New York, 1995.
18. S. Lang, *Linear Algebra*, 2nd edition, Addison-Wesley, 1970.



金昌洙(Chang Su Kim) 정회원

1971년 11월 12일생  
 1994년 2월 : 서울대학교 제어계측공학과 졸업(공학사)  
 1994년 3월~현재 : 서울대학교 제어계측공학과 석사과정 재학중



白勳(Hoon Paek) 정회원

1966년 7월 27일생  
 1990년 2월 : 서울대학교 제어계측공학과 졸업(공학사)  
 1992년 2월 : 서울대학교 대학원 제어계측공학과 졸업(공학석사)  
 1992년 3월~현재 : 서울대학교 대학원 제어계측공학과 박사과정 재학중

李商郁(Sang Uk Lee)

정회원

1949년 8월 11일생  
 1973년 2월 : 서울대학교 전기공학과 졸업(공학사)  
 1976년 5월 : 미국 Iowa 주립대학교 전기공학과 졸업(MSER)  
 1980년 1월 : 미국 Univ. of Southern California 전기공학과 졸업(Ph.D.)  
 1980년 1월~1981년 9월 : 미국 General Electric Co. 연구원  
 1981년 9월~1983년 3월 : 미국 M/A-COM. Research Center 선임연구원  
 1983년 4월~현재 : 서울대학교 제어계측공학과 교수