

고속 프랙탈 영상 부호화 기법

김 정 일[†] · 송 광 석^{††} · 강 경 인^{†††} · 박 경 배^{†††} ·
이 광 배[†] · 김 현 옥[†]

요 약

본 논문에서는 프랙탈 영상 부호화시 부호화 시간이 장시간 소요되는 단점을 보완하기 위한 고속 프랙탈 영상 부호화 알고리즘을 제안하고, 그 알고리즘의 성능을 기존의 방법과 비교 분석하였다. 기존의 프랙탈 영상부호화 방식은 원영상을 축소하여 비교될 영상으로 만들고, 축소된 영상에 대한 원영상의 축소변환의 고정점을 얻기 위해 축소된 영상의 전체 영역을 탐색하므로써 많은 부호화 시간이 소요되었다. 그러나, 제안한 알고리즘은 스케일링과 탐색영역제한 방식을 이용하여 부호화 시간을 대폭 단축시켰다. 그 결과로서 Jacquin 방법과의 비교시, 제안한 알고리즘은 최소 180배 이상의 부호화 시간을 단축시켰으며, 복원된 영상의 화질은 다소 감소하고 압축율은 약간 증가하였다. 따라서 제안한 알고리즘이 기존의 방법들에 비해 부호화 시간 면에서 크게 향상되었음을 확인할 수 있었다.

The Method of Fast Fractal Image Coding

Jeong Il Kim[†] · Kwang Suk Song^{††} · Kyeong In Kang^{†††} · Gyong Bae Park^{†††} ·
Kwang Bae Lee[†] · Hyen Ug Kim[†]

ABSTRACT

In this paper, we propose a fast fractal image coding algorithm to shorten long time to take on fractal image encoding. For its performance evaluation, the algorithm compares with other traditional fractal coding methods. In the traditional fractal image coding methods, an original image is contracted by a factor in order to make the corresponding image to be compared with. Then, the whole area of the contracted image is searched in order to find the fixed point of contractive transformation of the original image corresponding to the contracted image. It needs a lot of searching time on encoding. However, the proposed algorithm considerably reduces encoding time by using scaling method and limited search area method. On comparison of the proposed algorithm with Jacquin's method, the proposed algorithm is at least 180 times as fast as that of Jacquin's method on encoding time with a little degradation of the decoded image quality and a little increase of the compression rate. Therefore, it is found that the proposed algorithm largely improves the performance in the aspect of encoding time when compared with other fractal image coding methods.

1. 서 론

256 그레이 레벨(Gray level)을 가지는 256×256 크기의 영상을 1초에 30프레임을 전송할 경우 거의 15.7×10^6 비트/초의 데이터를 전송하여야 하며, 1프레임의 영상만을 대상으로 한다고 해도 대략 0.52×10^6 비트가 필요하다. 따라서, 이러한 방대한 양의 영상정

† 정 회 원: 명지대학교 전자공학과
†† 정 회 원: 한국전자통신연구소
††† 준 회 원: 명지대학교 전자공학과

논문접수: 1996년 1월 24일, 심사완료: 1996년 5월 15일

보를 전송 및 저장하기 위해서 영상 압축 기법의 개발이 필수 불가결한 요인으로 대두되어, 이와 관련된 많은 연구가 활발히 진행되어 왔다. 지금까지 연구되어온 영상 압축 기법에는 주로 변환 부호화(Transform coding), 벡터양자화(Vector quantization), 대역분할 부호화(Subband coding), 예측 부호화(Predictive coding), 엔트로피 부호화(Entropy coding), 웨이브렛 부호화(Wavelet coding), 프랙탈 부호화(Fractal coding)[1, 2, 3] 등이 있으며, 그 중에서도 프랙탈 부호화는 최근들어 더욱 활발하게 연구가 진행되고 있다.

프랙탈이라는 단어는 한 부분이 전체의 모습을 대표하는 형상이라는 뜻을 내포하고 있으며, 프랑스의 수학자인 Mandelbrot[4, 5]가 처음으로 프랙탈에 관한 이론을 정립시켰다. 그 이후 80년대에 Michael Barnsley[6, 7, 8]는 프랙탈 영상 부호화기법으로서 IFS(Iterative Function System)를 제시하여 높은 압축율로서 영상을 부호화하였으나, 수동적 방법으로 인하여 많은 시간이 요구되었다. Jacquin[9, 10, 11]은 기존의 수동적 방법에 의한 프랙탈 영상 부호화를 자동화하는 영상 부호화 알고리즘을 제시하였다. Jacquin의 방식은 원영상과 원영상을 1/2로 축소한 영상을 각각 같은 크기의 치역(Range) 블럭과 정의역(Domain) 블럭으로 분할한 후, 원영상의 치역 블럭과 가장 유사한 정의역 블럭을 1/2 축소 영상에서 찾았다. 이 방식에서는 많은 블럭을 찾기 위해 1/2로 축소된 영상의 전체 영역을 탐색하기 때문에 계산량이 증가하고 부호화 과정에 많은 시간이 소요되었다. 반면에 Monro [12]는 독립적인 블럭 단위로 부호화를 수행하는 방법을 제안하여 부호화 시간을 단축시켰으나, 그 방법은 블럭 단위가 고정되어 복원된 영상의 경계면에 불연속이 심하게 존재하여 영상 화질이 크게 떨어졌다.

본 논문에서 제안한 방식은 먼저, 스케일링 방식을 이용하여 원영상을 1/2과 1/4로 축소하고, 1/2 축소 영상과 1/4 축소 영상을 각각 4×4 크기의 치역 블럭과 정의역 블럭으로 분할한다. 그 다음, 1/2 영상의 치역 블럭과 가장 유사한 정의역 블럭을 1/4 축소된 영상에서 찾되, 1/4로 축소된 영상의 제한된 영역만을 탐색한다. 이 때, 대상 블럭이 윤곽선(Edge) 영역일 경우에는 2×2 크기의 블럭으로 분기하여 탐색을 수행한다.

본 논문은 다음과 같이 구성된다. 1장 서론에 이어

2장에서는 프랙탈 부호화의 이론적 배경에 대해서 간단하게 언급하고, 3장에서는 제안한 프랙탈 부호화 방법을 기술한다. 4장 실험결과 및 검토에서는 시물레이션 결과를 통해 제안한 방법과 기존의 방법을 비교·분석하고, 5장에서 결론을 맺는다.

2. 프랙탈 부호화의 이론적 배경

2.1 IFS 개념

IFS는 회전(Rotation), 스케일링(Scaling), 투영(Reflection) 및 평행이동(Translation)의 조합을 나타내는 유사변환(Affine transformation)으로 구성되며, 식 1)과 같이 표현된다.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = W \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \quad (2.1)$$

식 (2.1)은 평면상의 좌표 (x, y)가 계수 a, b, c, d, e, f 에 의해 새로운 점 (x', y')로 변환됨을 의미한다. a, b, c, d는 표1에 나타낸 바와 같이 8가지의 등장(等長) 변환(Isometry)을 나타내며, e, f는 평행이동을 나타낸다.

〈표 1〉 8방향 등장변환
〈Table 1〉 Eight directional isometry

Symmetry	Matrix	Description
0	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	동일
1	$\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$	y축에 대한 반사
2	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	x축에 대한 반사
3	$\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$	180° 회전
4	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	x = y축에 대한 반사
5	$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$	180° 회전
6	$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$	270° 회전
7	$\begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$	y = -x축에 대한 반사

IFS를 구성하는 유사 변환식들은 고정점(Fixed point)을 갖기 위해서 축소 변환(Contractive transformation)이어야 하며, 그 조건식은 식(2.2)와 같이 나타낼 수 있다.

$$d(W(A), W(B)) \leq s \cdot d(A, B) \quad (2.2)$$

$$\forall A, B \in U, 0 \leq s < 1$$

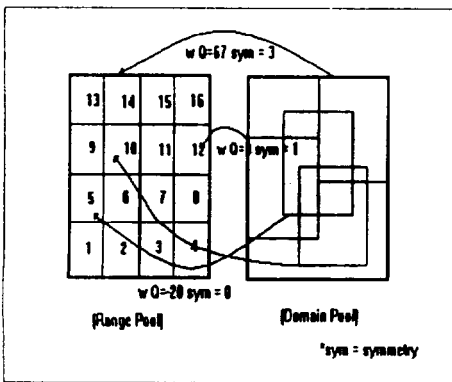
위식에서 U는 측도 공간(Metric space)을 나타내며, d는 Hausdorff 거리(distance)[6]를 나타낸다. 그리고 s는 변환식 w에 대한 축소율이다.

2.2 프랙탈 영상 부호화

영상은 화소의 값으로 인하여 2차원적 성격이 아닌 3차원적 성격을 가지므로 일반적인 유사 변환은 다음과 같이 3차원 모델로 표현할 수 있다.

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & P \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} R_x \\ R_y \\ Q_i \end{bmatrix} \quad (2.3)$$

위식에서 a_i, b_i, c_i, d_i 와 R_x, R_y 는 식(2.1)의 의미와 같으며, 계수 P는 0보다 크고 1보다 작은 값을 가지는 양수로써 Q_i 와 함께 변환된 화소 값을 나타낸다. 식(2.3)의 유사 변환식에 의해 원 영상의 치역 불릭이 1/2 축소 영상의 정의역 불릭과 비교연산에 의해 왜곡이 가장 적은 정의역 불릭을 수축변환점으로 선택하는 프랙탈 부호화 방법을 그림 1에 나타내었다. 그림에서 좌측 불릭은 치역을 나타내고, 우측불릭은 정의역을 나타내고 있다. wQ의 의미는 분할된 정의역과 치역 불릭간의 평균 화소값간의 차에 대한 수치를 나타내는 것이며, sym은 표 1에 나타낸 바와 같이 8가지의 대칭변환중 가장 적합한 변환의 번호를 의미한다.



(그림 1) 프랙탈 부호화 (Fig. 1) Fractal coding

다. 그림 1의 맨 위에 나타낸 $wQ=67 \text{ sym}=3$ 에서 $wQ=67$ 은 해당 정의역 불릭과 치역 불릭 14간의 평균 화소값의 차가 67임을 의미하며, $\text{sym}=3$ 은 대칭 변환 3번 즉, 180°회전을 의미하는 것이다. 따라서, 치역 불릭 14는 대칭 변환 3번을 적용해서 가장 적은 왜곡 값을 나타내는 해당 정의역 불릭으로 부호화되는 것이다.

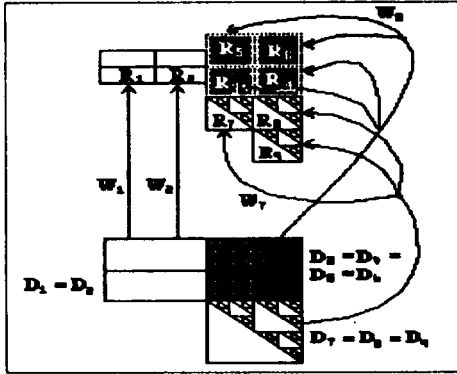
3. 제안한 프랙탈 영상 부호화 방법

3.1 탐색 영역의 제한

기존의 프랙탈 부호화 방법에서는 원 영상의 치역 불릭이 축소 변환되는 점을 찾기 위해서 축소된 영상의 정의역 불릭들과 비교[9, 10, 11]하였다. 이 때, 축소변환의 고정점을 찾기 위해서는 축소된 영상의 전체 정의역 불릭들에 대해 비교가 되기 때문에 상당량의 연산 후 원영상의 치역 불릭에 대해 최소의 에러를 나타내는 대응 정의역 불릭을 찾아 부호화하였다. 그 경우 대부분의 256×256 크기의 영상 부호화시 486 DX/2 66MHz 컴퓨터를 사용하여 최소한 10시간 이상의 연산을 필요로 하였다. 그러므로 부호화 시간을 효율적으로 줄이기 위해서 Jacquin은 불릭 분류 방식을 사용하여 축소된 영상에 대한 탐색시간 및 비교 연산 시간을 줄였으며[9, 10, 11], Monro는 윤곽선 불릭에 대해 그 불릭을 4등분하고 4등분된 부불릭들에 대해서만 탐색을 수행하였다[12]. 또한 스케일링 방식에서는 치역 불릭과 정의역 불릭의 크기를 축소 시킴으로써 비교 연산 시간을 단축시켰다.

원 영상의 축소 변환점은 축소된 영상의 좌표주위이며 따라서 탐색영역을 그 좌표 주위로 한정한다면 탐색영역을 줄일 수가 있음을 그림 2에서 잘 보여주고 있다. 그림 2에서는 축소되지 않은 흑백 원영상에 대한 정의역 불릭을 축소 변환시켜 그에 대응하는 원영상의 치역 불릭을 찾는 방법을 도시하였다. 그림에서 상단에 위치한 불릭들은 치역 불릭에 해당하며, 하단에 위치한 불릭들은 치역 불릭에 해당한다. W는 축소 변환을 의미하는 것으로 Collage 이론을 나타내고 있으며, 각각의 D(정의역) 불릭을 축소 변환하여 각각의 R(치역) 불릭을 구할 수 있음을 나타낸다. 표 2에는 그림 2에 대한 IFS 코드를 나타내었다. 표 2의 결과는 원 영상의 치역 불릭에 대한 축소 변환점은

바로 이웃한 원영상의 정의역 블럭에서 찾을 수 있을
을 보여준다[6].



(그림 2) 치역 블럭 R_i와 정의역 블럭 D_i의 대응
(Fig. 2) Relationship between range block R_i and domain block D_i

<표 2> 그림 2에 대한 IFS 코드
<Table 2> IFS codes for Fgi. 2

Map#	Rx	Ry	Dx	Dy	Symmetry
1	0	2.5	0	2	0
2	1	2.5	0	2	0
3	2	2	2	2	0
4	3	2	2	2	0
5	2	3	2	2	0
6	3	3	2	2	0
7	2	1	2	0	0
8	3	1	2	0	0
9	3	0	2	0	0

위의 이론 즉, 2가지의 화소값만을 가지는 흑백 영
상에 대한 프랙탈 변환에 대한 이론은 256가지의 화
소값을 가지는 그레이스케일(Grayscale) 영상[6]에 대
해서도 적용될 수 있다. 그레이스케일 영상의 경우 8
× 8블럭단위로 그 주변에 한정하여 탐색을 할 경우,
정확한 축소 변환점을 찾지 못하는 경우가 존재한다.
이것은 영상이 화소값을 가짐으로 흑백영상과 다른
이유이다. 선택되어진 블럭이 단순한 화소의 분포를
보인다면 흑백영상과 같은 원리로 그 주변에서만 축
소 변환점을 찾는 것이 가능하나, 선택되어진 블럭이

다양한 화소값을 갖는다면 축소된 영상의 주변블럭
들에 대해서만 축소변환점을 찾는 것은 무리이다. 따
라서 선택되어진 블럭이 다양한 분포의 화소값을 갖
는 윤곽선 블럭인 경우는 4개의 블럭으로 세분하여
축소영상의 주변 블럭에서 탐색한다면 상대적으로
화소 분포가 줄어들어 축소 변환점을 찾는 것이 가능
하다.

본 논문에서는 먼저 스케일링 방식을 이용하여 원
영상을 1/2과 1/4로 축소한 후, 1/2 축소 영상과 1/4
축소 영상을 이용한다. 이러한 축소 영상을 그림 3에
나타내었다. 1/2 축소 영상의 치역블럭과 가장 유사
한 정의역 블럭을 1/4 축소 영상에서 찾되, 그림 4와
같은 방법으로 탐색영역을 제한함으로써 부호화 시

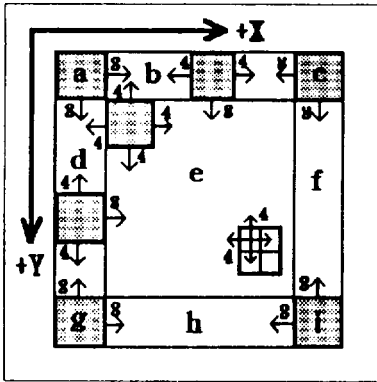


(a)



(b)

(그림 3) 논문에서 사용된 축소 영상
(a) 1/2 축소영상 (b) 1/4 축소영상
(Fig. 3) Contracted image used in this paper
(a) 1/2 contracted image (b) 1/4 contracted image



(그림 4) 위치에 따른 탐색영역 분류
(Fig. 4) Positional classification of search area

간을 줄였다. 그림 4에서 보는 바와 같이 축소 영상은 9개의 영역으로 분리를 할 수가 있다. 각 영역에 따라 탐색 영역도 각기 다르게 진행된다. a는 x, y축의 양의 방향으로 각각 8화소의 영역만을 조사하며, c는 x축의 음의 방향으로 8화소 영역 그리고 y축의 양의 방향으로 8화소 영역을 g는 x축의 음의 방향 8화소 영역 그리고 y축의 양의 방향 8화소 영역을 조사하고, i의 탐색 영역은 x, y축의 음의 방향으로 각각 탐색을 한다. b의 탐색영역은 x축의 음의 방향으로 4화소 영역 양의 방향으로 4화소 영역 그리고 y축의 양의 방향으로 8화소 영역을 탐색하게 되고, d는 y축의 음의 방향 4화소 영역 양의 방향으로 4화소 영역 그리고 x축의 양의 방향으로 8화소 영역을 조사한다. f의 탐색 영역은 x축의 음의 방향으로 8화소 영역을 탐색하고 y축의 음과 양의 방향으로 4화소씩 각각 조사하게 된다. h는 반대로 y축의 음의 방향 8화소 영역을 조사하며 x축의 음과 양의 방향으로 4화소씩 각각 조사하게 된다. 마지막으로 중앙 부분의 e 영역은 x, y축의 음과 양의 방향으로 각각 4화소 영역을 1화소씩 움직이며 탐색을 하게 된다. 물론 지금까지 설명한 탐색 방법에는 대각선 방향도 포함된다. 만약 위의 각 영역에서 선택되어진 블럭이 화소의 활동도가 높은 윤곽선 블럭인 경우는 2x2 블럭으로 분기하여 각 영역에 해당되는 탐색영역을 조사하게 된다. 따라서 본 논문에서 사용된 알고리즘은 축소 영역이 국부적으로 제한되어 부호화 시간이 상대적으로 극히 짧

아짐을 알 수 있다. 또한 본 논문에서 단순 블럭과 윤곽선 블럭을 분류하기 위해 사용한 블럭 분류기는 다음 수식과 같다.

$$\sum_{i=1}^{B^2-1} (P_i - P_{i+1}) \geq Th \quad (3.1)$$

위식에서 B는 모(Parent) 블럭과 자(Child) 블럭의 크기를 나타내고, P는 각 블럭내의 화소값을 나타낸다. 따라서, 식 (3.1)의 의미는 인접 화소값간의 차의 합이 임계치(Threshold) Th 값보다 크거나 같으면, 윤곽선 블럭으로 분류되고, 그 미만이 되면, 단순 블럭으로 분류한다. 그 다음, 윤곽선 블럭으로 분류된 모블럭은 다시 B/2의 크기를 갖는 4개의 동일 크기의 자블럭으로 분할된다. 이렇게 분할된 4개의 자블럭에 대해서도 동일한 방법으로 단순 블럭과 윤곽선 블럭을 분류한다.

3.2 비트 할당

비트 할당은 단순 영역과 윤곽선 영역으로 구분하여 처리하였다. 단순 영역의 경우에는 블럭의 형태를 표현하기 위한 1 비트와 단위 블럭간의 평균 화소값의 차를 표현하기 위한 9 비트, 그리고 좌표값에 6비트가 필요하므로 총 16 비트를 할당한다. 윤곽선 영역의 경우에는 블럭의 형태와 평균 화소값의 차뿐만 아니라, 좌표값과 등장(等長)변환(Isometry) 값이 필요하므로 총 19 비트를 할당한다. 평균 화소값의 차에는 부호 비트가 포함된 것이다. 이것을 표 3에 나타낸다.

<표 3> 비트 할당
<Table 3> Bits allocation

영역	구분	세부 할당(bits)	총 할당 (bits)
단순 영역	블럭 형태	1	16
	평균 화소값의 차	9	
	좌표값	6(3+3)	
윤곽선 영역	블럭 형태	1	19
	평균 화소값의 차	9	
	좌표값	6(3+3)	
	등장 변환값	3	

비트 계산은 다음과 같다. 먼저, 영상을 두가지 형태의 블럭 즉, 단순영역 블럭과 윤곽선 영역 블럭의 정보를 표현하는데 소요되는 비트를 각각 I_s, I_e 로 나타내고, 영상에서 분류된 단순 블럭과 윤곽선 블럭의 총 갯수를 각각 N_s, N_e 로 나타내도록 한다. 그리고, 영상에서 최초로 분할한 블럭 즉, 모블럭의 크기를 $B \times B$ 라 하고, 윤곽선 블럭인 경우에 다시 한번 분할한 블럭 즉, 자블럭의 크기를 $B/2 \times B/2$ 라 할 때, 비트율의 계산은 다음식과 같다.

$$\frac{N_p I_c + N_s I_s + N_e I_e}{N_p B^2} \text{ bpb(bit/pixel)} \quad (3.2)$$

위 식에서 I_c 는 모블럭에서 자블럭의 분할에 대한 배열을 나타내는데 필요한 4비트가 소요된다. 다시 말해서, 그러한 경우는 모두 12가지-1 parent no child의 한가지 경우, 1 parent 1 child의 네가지 경우, 1 parent 2 children의 여섯가지 경우, no parent 4 children의 한 가지 경우-가 존재하게 된다. 따라서, 12가지의 배열을 표현하기 위해서는 4비트가 필요하다. N_p 는 영상에서 모블럭의 총 갯수를 나타낸다.

3.3 복호화

전송되어진 프레임당 부호화 값을 이용하여 블럭의 형태에 따라 복호화를 진행한다. 블럭의 형태가 단순영역인 경우는 4×4 블럭단위로 복호화가 진행되며, 윤곽선 블럭인 경우 2×2 블럭단위로 복호화가 진행

된다. 물론, 이러한 복호화는 반복(Iteration)을 통하여 진행되는데 통상 8회 이상의 반복 이후에는 PSNR의 차이가 거의 없기 때문에 반복 횟수를 8회로 제한하였다.

3.4 화질 평가

재구성된 영상의 화질을 평가하는 방법으로 RMSE (Root Mean Square Error)와 PSNR(Peak Signal to Reconstruction Noise Error)을 사용하였다[14]. $B \times B$ 영상을 A라 하고, 복원된 영상을 \hat{A} 라 표시하면, RMSE는 다음과 같은 식 (3.3)으로 표시된다.

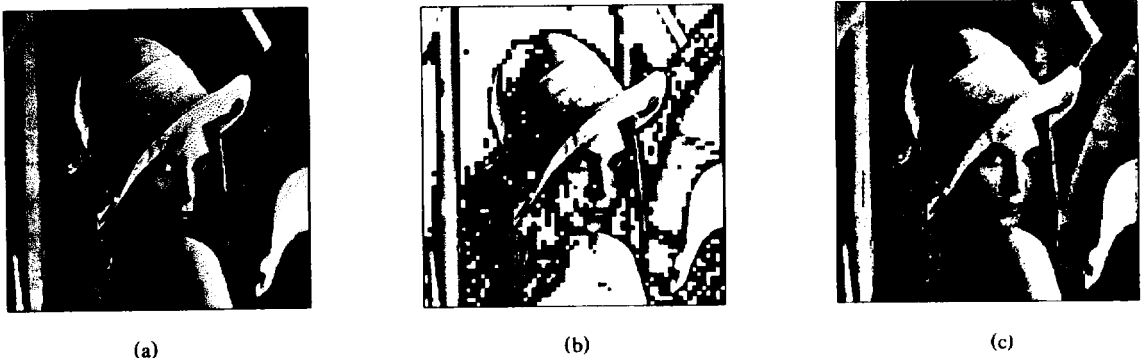
$$RMSE = \sqrt{\frac{1}{B^2} \sum_{i=1}^B \sum_{j=1}^B (A(i, j) - \hat{A}(i, j))^2} \quad (3.3)$$

RMSE를 이용한 PSNR(dB)의 측정은 8비트(0~255) 영상에 대해 다음과 같은 식 (3.4)를 이용하여 계산하였다.

$$PSNR = 20 \log_{10} \left(\frac{255}{RMSE} \right) \quad (3.4)$$

IV. 실험결과 및 검토

실험에 사용된 영상은 256×256 크기의 Lenna, Baboon, Couple, Cronk, Girl, House, Jaguar, Miss America, Peppers, Salesman, Woman의 11가지 표준



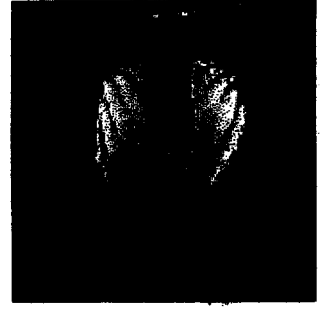
(그림 5) Lenna 영상
 (a) 원영상 (b) 분류된 단순영역 (c) 복원된 영상
 (Fig. 5) Lenna image
 (a) The original image (b) The classified shade block (c) The decoded image



(a)



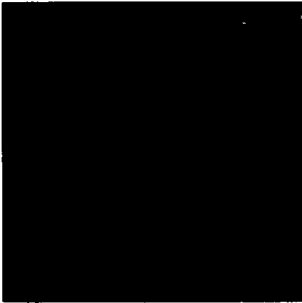
(b)



(c)

(그림 6) Baboon 영상
(a) 원영상 (b) 분류된 단순영역 (c) 복원된 영상
(Fig. 6) Baboon image

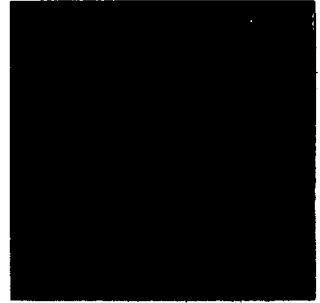
(a) The original image (b) The classified shade block (c) The decoded image



(a)



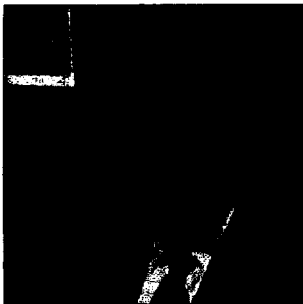
(b)



(c)

(그림 7) Couple 영상
(a) 원영상 (b) 분류된 단순영역 (c) 복원된 영상
(Fig. 7) Couple image

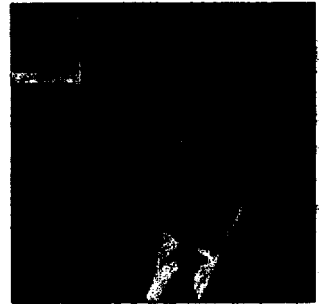
(a) The original image (b) The classified shade block (c) The decoded image



(a)



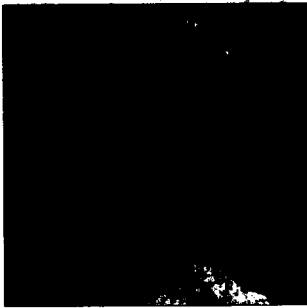
(b)



(c)

(그림 8) Cronk 영상
(a) 원영상 (b) 분류된 단순영역 (c) 복원된 영상
(Fig. 8) Cronk image

(a) The original image (b) The classified shade block (c) The decoded image



(a)



(b)



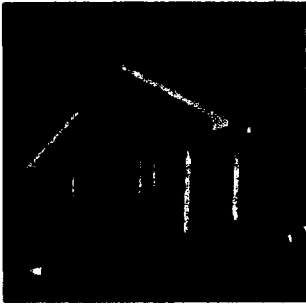
(c)

(그림 9) Girl 영상

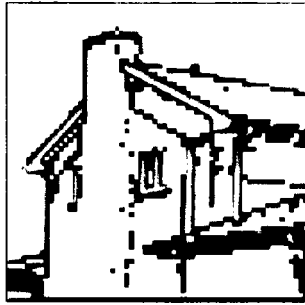
(a) 원영상 (b) 분류된 단순영역 (c) 복원된 영상

(Fig. 9) Girl image

(a) The original image (b) The classified shade block (c) The decoded image



(a)



(b)



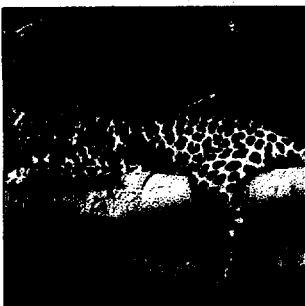
(c)

(그림 10) House 영상

(a) 원영상 (b) 분류된 단순영역 (c) 복원된 영상

(Fig. 10) House image

(a) The original image (b) The classified shade block (c) The decoded image



(a)



(b)



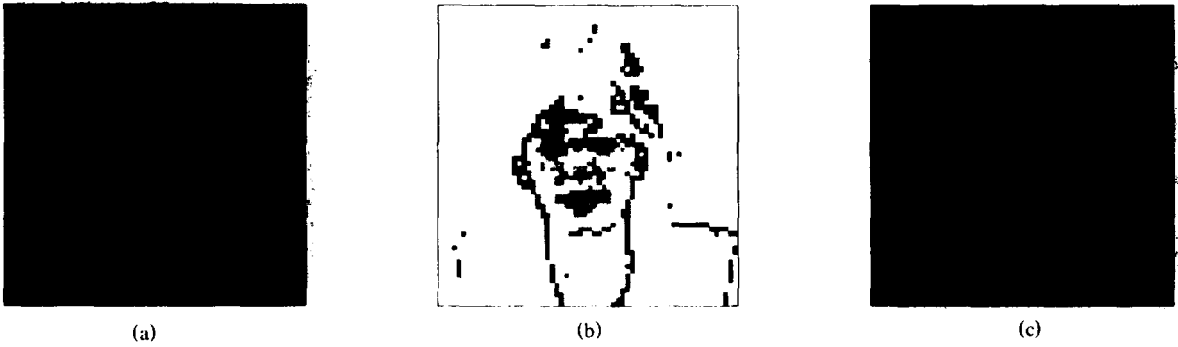
(c)

(그림 11) Jaguar 영상

(a) 원영상 (b) 분류된 단순영역 (c) 복원된 영상

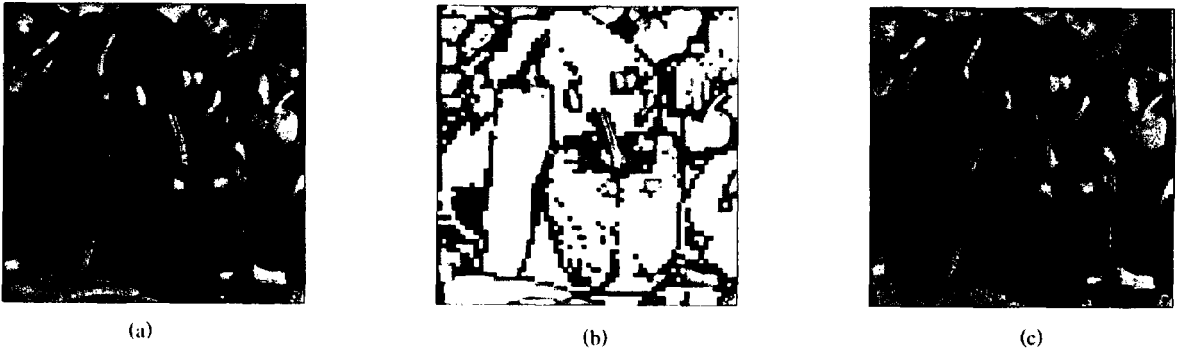
(Fig. 11) Jaguar image

(a) The original image (b) The classified shade block (c) The decoded image



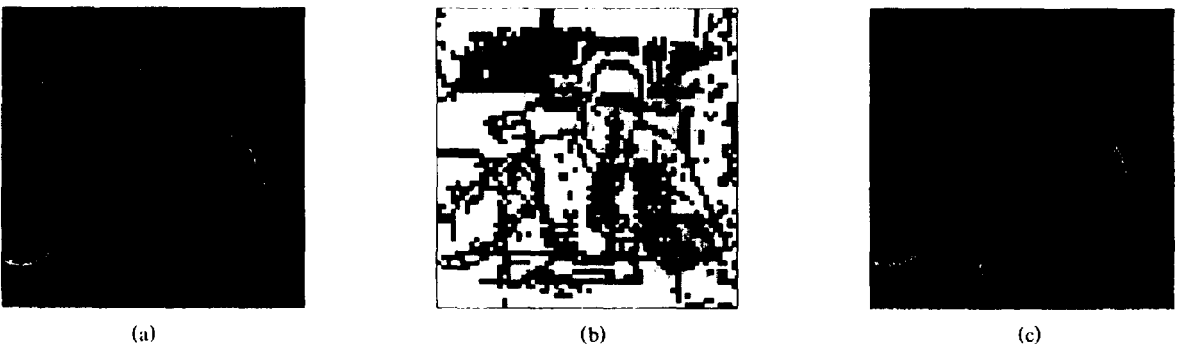
(그림 12) Miss America 영상
(a) 원영상 (b) 분류된 단순영역 (c) 복원된 영상
(Fig. 12) Miss America image

(a) The original image (b) The classified shade block (c) The decoded image



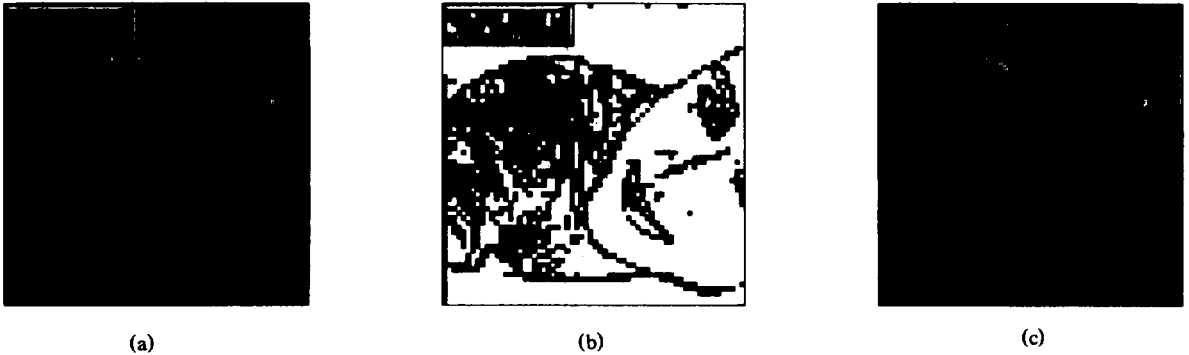
(그림 13) Peppers 영상
(a) 원영상 (b) 분류된 단순영역 (c) 복원된 영상
(Fig. 13) Peppers image

(a) The original image (b) The classified shade block (c) The decoded image



(그림 14) Salesman 영상
(a) 원영상 (b) 분류된 단순영역 (c) 복원된 영상
(Fig. 14) Salesman image

(a) The original image (b) The classified shade block (c) The decoded image



(그림 15) Woman 영상
 (a) 원영상 (b) 분류된 단순영역 (c) 복원된 영상
 (Fig. 15) Woman image

(a) The original image (b) The classified shade block (c) The decoded image

<표 4> 제안한 방법과 스케일링 및 Jacquin 방법의 성능 비교

<Table 4> Performance comparison between proposed method, scaling method and Jacquin method

영상	제안된 방법			스케일링 방법			Jacquin 방법		
	부호화 시 간	압축율 (bpp)	PSNR (dB)	부호화 시 간	압축율 (bpp)	PSNR (dB)	부호화 시 간	압축율 (bpp)	PSNR (dB)
Lenna	58초	0.72	24.46	21분 08초	0.83	24.08	05시간 44분	0.65	27.24
Baboon	64초	0.86	23.14	46분 18초	0.97	23.79	14시간 22분	1.52	26.21
Couple	35초	0.47	25.72	17분 32초	0.59	29.01	05시간 59분	1.11	32.17
Cronk	31초	0.40	27.36	08분 44초	0.51	31.78	01시간 57분	0.56	34.76
Girl	35초	0.46	27.19	13분 26초	0.46	29.62	03시간 54분	0.82	32.87
House	31초	0.40	28.49	05분 55초	0.46	30.05	01시간 36분	0.29	34.03
Jaguar	63초	0.76	21.54	36분 20초	1.04	20.59	09시간 45분	1.39	25.18
Miss America	25초	0.32	29.31	05분 51초	0.44	34.72	02시간 33분	0.51	37.04
Peppers	45초	0.58	27.36	12분 49초	0.61	28.16	03시간 08분	0.85	32.57
Salesman	40초	0.56	26.40	16분 58초	0.70	22.47	05시간 42분	1.16	33.09
Woman	43초	0.56	24.92	17분 46초	0.69	25.68	04시간 57분	1.02	29.56

영상이다. 이들 영상을 그림 5부터 그림 15까지의 (a)에 나타내었다. 시뮬레이션은 486 DX/2 66MHz 컴퓨터를 이용하였으며, 시뮬레이션의 결과를 그림 5부터 그림 15까지의 (b)와 (c)에 각각 나타내었다. 각 그림의 (b)는 분류기(Classifier)를 통해 분류된 단순영역을 표시한 영상을 나타낸 것이고, (c)는 복원된 영상을 각각 나타낸 것이다. 또한 제안한 방법과 스케일링

방법 그리고 Jacquin 방법간의 성능 비교를 표 5에 나타내었다. 표에서 보는 바와 같이 부호화 시간은 Jacquin 방법의 경우 수십시간대가 소요되고, 스케일링 방법의 경우 수십분대가 소요되는 것을 알 수 있다. 그러나 제안한 방법은 부호화에 소요되는 시간은 불과 수십초대인 것을 확인할 수 있다. 압축율 면에서도 세가지 방법 중에서 제안한 방법이 가장 효율적인

것을 알 수 있다. PSNR은 제안한 방법이 Jacquin 방법에 비해 약간 떨어지기는 하나 큰 차이는 없으며, 그림 5부터 그림 15까지의 영상을 비교해 보더라도 시각적인 차이는 거의 느낄 수 없다.

5. 결 론

본 논문에서는 부호화 시간을 최소로 줄이기 위한 고속 프랙탈 부호화 알고리즘을 제안하였다. 먼저, 기존에 제시된 프랙탈 부호화 방식인 Jacquin 방식과 스케일링 방식에 대해 설명하고, 그들에 대한 문제점을 제시하였다. 그리고 제시된 문제점을 해결하기 위해 스케일링 방식과 탐색영역제한 방식을 제안하여 시뮬레이션함으로써 한층 발전된 결과를 얻을 수 있었다. 세가지 방식에 대한 비교 결과, 부호화에 소요되는 시간은 Jacquin 방식의 경우 수십시간대에 이르고, 스케일링 방식의 경우는 수십분대가 소요되나, 제안한 방식의 경우에는 수십초대로 감소된 것을 확인할 수 있었다. 압축율면에서도 다른 두 가지 방식에 비해 제안한 방식이 약간 향상되었음을 알 수 있었다. 그러나, 부호화 시간과 압축율을 향상시키기 위해 약간의 PSNR의 희생을 감수하였으나 좀 더 정교한 분류기를 사용한다면 떨어진 화질을 향상시킬 수 있을 것이다. 따라서, 향후 더욱 효율적인 분류기에 대한 연구가 필요할 것으로 판단되며, 본 논문에서 제안한 알고리즘을 동영상 전송에 적용한다면 좋은 결과가 있을 것으로 기대된다.

참 고 문 헌

[1] M.Rabbani and P.W.Jones, *Digital Image Compression Techniques*, SPIE Optical Engineering Press, 1991.
 [2] B.J.Thompson, *Image Coding and Compression*, SPIE Optical Engineering Press, 1992.
 [3] Y.Fisher, *Fractal Image Compression Theory and Application*, SPIE Optical Engineering Press, 1995.
 [4] H.O.Peitgen, H.Jürgens and D.Saupe, *Chaos and Fractals*, Springer-Verlag, New York, Inc. 1992.
 [5] H.O.Peitgen, H.Jürgens and D.Saupe, *Fractals*

for the Classroom, Springer-Verlag, New York, Inc. 1991.

[6] M.F.Barnsley and L.P.Hurd, *Fractal Image Compression*, AK Peters, Wellesley, Massachusetts, 1992.
 [7] M.F.Barnsley and S. Demko, "Iterated function systems and the global construction of fractals," in Proc. of Royal Soc. London Vol. A399, pp. 243-275. 1985.
 [8] M.F.Barnsley, *Fractals Everywhere*, Academic Press, Inc 1988.
 [9] A.E.Jacquin, "A fractal theory of iterated Markov operators with applications to digital image coding," Ph.D. dissertation, Georgia Tech, 1989.
 [10] A.E.Jacquin, "A novel fractal block-coding technique for digital Image," ICASSP 90 Proceedings.
 [11] A.E.Jacquin, "Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations," IEEE Trans. on Image Process., Vol. 1 No 1, Jan. 1992.
 [12] D.M.Monro and F.Dudbridge, "Fractal approximation of image blocks," ICASSP pp. 485~488, 1992.
 [13] 안영훈, "프랙탈 스케일링 방식을 이용한 영상 부호화," 명지대학교 석사학위논문, 1994.
 [14] J.R.Deller, JR.J.G.Proakis and J.H.L.Hansen, *Discrete-Time Processing of Speech Signals*, Macmillan Publishing Company, New York, 1993.



김 정 일

1989년 명지대학교 전자공학과 졸업(공학사)
 1994년 8월 명지대학교 전자공학과 졸업(공학석사)
 1995년 3월~현재 명지대학교 전자공학과 박사과정 재학중

관심 분야: 영상통신, 음성통신, 멀티미디어 시스템

송 광 석

1979년 고려대학교 전자공학과 졸업(공학사)
 1981년 고려대학교 대학원 전자공학과 졸업(공학석사)
 1992년 고려대학교 대학원 전자공학과 졸업(공학박사)
 1992년 Georgia Institute of Technology 객원 연구원
 1982년~현재 한국 전자통신연구소 제어시스템 연구실장

관심 분야: Fault Tolerant Control System, Computer Architecture, ATM Switching System



강 경 인

1994년 명지대학교 전자공학과 졸업(공학사)
 1996년 2월 명지대학교 전자공학과 졸업(공학석사)
 1996년 3월~현재 명지대학교 전자공학과 박사과정 재학중

관심 분야: 멀티미디어(영상 통신, 음성통신)

박 경 배

1994년 명지대학교 전자공학과 졸업(공학사)
 1996년 2월 명지대학교 전자공학과 졸업(공학석사)
 1996년 3월~현재 명지대학교 전자공학과 박사과정 재학중

관심 분야: 멀티미디어(영상 통신, 음성통신)

이 광 배



1979년 고려대학교 전자공학과 졸업(공학사)
 1979년~1981년 고려대학교 전자공학과 졸업(공학석사)
 1981년 3월~1982년 3월 삼성반도체연구소
 1982년 3월~1983년 4월 금성연구소

1984년~1986년 Univ. of Southern California, Computer Engineering 전공(공학석사)

1986년~1991년 Arizona state Univ., Electrical Engineering 전공(공학박사)

1992년~현재 명지대학교 전자공학과 조교수

1994년~현재 대한 전자공학회 논문편집위원

관심분야: 멀티미디어(영상 및 음성 신호처리), 병렬처리 및 고속 컴퓨터(prolog 방식), Communication System(고장 감내형).

김 현 옥



1978년 2월 고려대학교 전자공학과 졸업(공학사)

1980년 2월 고려대학교 전자공학과 졸업(공학석사)

1987년 2월 고려대학교 전자공학과 졸업(공학박사)

1980년 3월~1981년 2월 동양공업 전문대학 전자과 전임강사

1881년 3월~1988년 1월 명지대학교 전자공학과 교수

1988년 2월~1990년 1월 Dept. of Computer Science of Arizona State University Adjunct Faculty

1990년 3월~현재 명지대학교 전자공학과 교수

관심분야: 병렬처리 컴퓨터 시스템, 고장 감내 시스템, 영상통신, 음성통신