

## 세그먼트기법을 이용한 프랙탈 영상 부호화에 대한 연구

서 주 하 · 최 황 규 · 조 철 회

### A study On the Image Coding Based on the Segmented Fractal Coding

Ju-Ha Seo · Hwang-Kyu Choi · Churl-Hee Cho

---

#### ABSTRACT

Fractal coding is a promising method for image compression, but it has not lived up to its promise as low bit-rate image compression scheme. The existing algorithms for finding self-mapping contractive transforms are computationally expensive and offer a poor rate-quality tradeoff. In this paper, we propose a segment based fractal coding. We classify the range blocks into shade, midrange or edge blocks, and segment edge block along the edge. And we apply midrange coding scheme for each segment. Our experiments show that our method gives better rate-quality tradeoff than current fractal block coding methods.

---

#### 1. 서 론

문자, 그래픽, 정지화상, 오디오 및 비디오 데이터를 결합하여 표현하는 멀티미디어 데이터 중에서 비디오데이터는 그 양이 방대하고 동시성을 가지므로 처리하기 어려운 반면 전달효과가 크다는 면에서 멀티미디어 서비스의 핵심이라 할 수 있다. 따라서 이를 통신망을 통하여 서비스하려면 전송 및 저장에 압축기법의 사용이 필수적이다. 따라서 ISO/IEC 의 MPEG 그룹에

의하여 비디오 압축에 대한 표준화가 활발히 진행되고 있다. 지금까지 동화상에 대하여는 64K x p의 전송율로 화상회의 등에 실시간 동화상 부호화 및 복호화를 위한 H.261[1], 1.5 Mbps 급의 전송율로 저장 매체를 위한 MPEG-1[2] 및 15 Mbps의 전송율로 방송 수준의 화질을 전송하기 위한 MPEG-2[3]의 표준화가 진행되고 있다. 그리고 최근에 기존의 전화망을 사용하여 10 KHz의 전송율로 동화상을 전송할 수 있는 초저속 전송율에서의 동화상 서비스에 대한 표준화 작업이 MPEG-4[4]라 하여 진행되고 있다. 현재 통신망으로서 가장 널리 분포하는 것은 공중전화망이며, 이를 이용한 동화상 전송을 위한 압축방법은 멀티미

---

강원대학교 컴퓨터공학과 부교수

\* 강원대학교 컴퓨터공학과 조교수

\*\* 강원대학교 전자공학과 박사과정

디어 서비스를 앞 당길 것으로 보며 이에 대한 표준화 일정은 1998년 이후에 표준화가 완성될 것으로 본다.

MPEG-4에서는 기존에 DCT를 기반으로 하고 움직임 보상과 엔트로피 코딩을 사용한 H.261, MPEG-1 및 MPEG-2와는 다른 압축방법을 사용하여야 낮은 전송율에서 동화상 서비스를 할 수 있으므로, 웨이브릿, 객체 기반, 모델 기반 및 프랙탈 기반의 방법이 거론되고 있다. 이중 웨이브릿 변환[5]은 푸리에 변환 처럼 기저함수들의 집합으로 분해하여 표현하는 방법이다. 모델기반 기법[6]은 대상 물체에 대한 모델과 지식을 송수신 양측에 준비하고, 송수신측에서 입력 영상을 분석하고 수신측은 분석결과에 따른 결과를 가지고 동영상 합성하는 방법이다. 그리고 객체기반 압축 방법[7]은 각 영상 프레임을 영상분석을 통하여 움직임이 없는 배경과 3차원의 움직임 객체로 구분하여 움직임 객체들의 모양, 색정보 및 움직임 정보에 대한 파라미터를 추출하여 전송하는 방법이다. 여기서 모양은 객체의 윤곽 정보이고, 색정보는 객체 영역의 질감을 말한다.

프랙탈 압축기법[8]은 영상의 자기유사성을 이용하여 영상내의 각 블록을 유사한 부분을 찾아 축소변환하는 파라미터로 부호화한다. 이 기법은 이론적으로 압축율이 높아 많은 연구가 행하여지고 있다.

Jacquin[9]과 Fisher[10]는 블록들을 분류기에 거쳐 이미지의 복잡도에 따라 코딩 방법을 달리했으며, 최소자승법으로 가장 적합한 축소변환의 계수들을 계산하여 IFS 부호들을 얻었다. 그러나, 원 이미지로부터 중복을 허용하며 얻은 모든 정의역블럭(domain block)과 비교하여 치역블럭(range block)과 가장 비슷한 정의역블럭을 찾기 위해 많은 시간이 소비되었다. 반면, Monro[11]는 부호화하고자하는 치역블럭

을 포함한 영역을 정의역블럭으로 하여 고차식의 계수들을 계산한다. Jacquin 알고리즘은 치역 블럭과 축소변환한 정의역블럭과 가장 가까운 것을 찾기 위해 너무 많은 시간이 소모되는 반면에 Monro알고리즘은 한개의 정의역블럭에서 고차식으로 표현된 축소 변환을 계산하므로 계산 시간이 줄어 실시간 부호화에 더 유리하다. 이 변환은 BFT(Bath Fractal Transform)라고 부른다. 비트율과 복원된 이미지의 질을 효과적으로 조정하기 위하여 블럭의 복잡도에 따라 이에 적합한 BFT를 선택해서 효율적으로 압축하였다.

본 논문은 코딩 속도와 높은 압축을 하기위해서, 프랙탈 기법과 세그먼트 기법을 혼합한 기법을 제안한다. 세그먼트 기법은 블럭내에 에지(edge)를 따라 몇개의 다각형 영역으로 분할하여 각 영역에 프랙탈 부호화 기법을 적용한다. 본 논문에서는 다각형의 윤곽을 표현하는 윤곽선 정보를 줄이기 위해 다각형의 수를 2 이하로 제한하였다.

이 논문은 2장에서 영상 부호화에 프랙탈 기법을 처음 도입한 Jacquin의 방법을 소개하고, 3장에서는 본 논문에서 제안한 세그먼트기법을 이용한 프랙탈 코딩에 대한 설명과 4장에서는 이에 대한 시뮬레이션 결과에 대하여 기술하였다.

## 2. 프랙탈 영상 코딩

Jacquin 알고리즘은 이미지의 자기유사성을 이용하여 원래의 영상을 블럭으로 분할한 후 한 블럭식을 프랙탈 이론에 기초를 둔 축소변환의 파라미터로 표현함으로써 영상을 압축하는 방법이다. 이는 자기 변환(Self-transformality)이라는 프랙탈 이론에 근거한, 이미지의 한 부분은 전체를 적당히 변환하여 근사화 시킬수 있는 것, 변환에

바탕을 두고 있다. 실제로 Tree, Fern, Clouds[12] 등과 같은 프랙탈 영상은 자기 변환 성질을 이용하여 몇개의 변환으로 영상을 만들 수 있으나, 실제의 자연 영상은 이런 성질을 원영상전체에서 추출하기 어려우므로 작은 블록으로 나누어 각각의 변환을 구한다. 영상의 복원은 모든 블록의 변환식들을 이용하여 이루어 진다.

### 2.1 치역 블록(Range Block)

$N \times N$  이미지를 압축할 때, 보통  $n \times n$ 으로 구성된 기본적인 블록으로 분할한다. 이러한 블록의 분할 방법은 JPEG 등과 같은 영상을 압축하는 방법 등에서 사용했으며, 분할된 블록들은 이웃한 블록과 겹치지 않게 이루어졌기 때문에,  $(N/n)^2$ 개의 블록들이 얻어진다. 이 블록들을 치역 블록이라고 부르는데, 이러한 명칭은 유사 변환의 치역에 해당되기 때문이다. 이 치역 블록은 코딩의 기본적인 단위이며, 보통 그 크기는  $4 \times 4$ ,  $8 \times 8$ ,  $16 \times 16$ 으로 2의 제곱승으로 사용될 수 있으나 프랙탈 블록 코딩으로 영상을 압축할 때는  $4 \times 4$ 나  $8 \times 8$ 이 적합한 크기로 사용되었다.

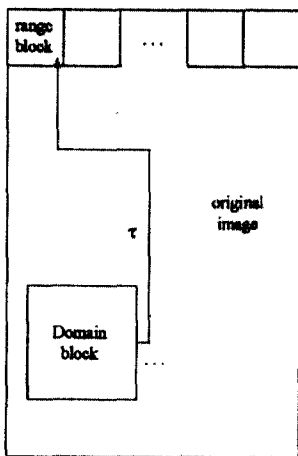


Fig. 1. Fractal Block Coding

### 2.2 정의역 블록(Domain Block)

정의역 블록은 이에 유사 변환을 적용시켜 부호화할 치역 블록과 유사하고, 그 크기가 치역 블록 크기 보다 큰 블록을 말하며, 계산의 비용을 줄이기 위하여 보통 정의역 블록은 치역 블록의 두배인 크기를 사용한다. 이런 정의역 블록들의 모음을 domain pool이라고 한다. 프랙탈 블록 부호화 기법을 사용할 때 모든 정의역 블록들을 축소변환하고, 그 결과중 치역 블록과의 차이가 가장 작은 블록을 찾아내야 하기 때문에 계산시간은 정의역 블록의 갯수와 상당한 관련이 있다. 그러나, 원 영상으로부터 되도록 많은 블록을 얻어야 다양한 정의역 블록으로 domain pool를 구성할 수 있고, 다양한 정의역 블록이 있어야 원영상에 가까운 축소 변환을 얻을 수 있다. 따라서 다양한 정의역 블록을 얻기 위해 원래의 영상을 중복을 허용하면서 분할한다.

### 2.3 유사 변환(Affine Transform)

프랙탈 부호화에서의 축소변환의 가장 간단한 형태는 다음과 같은 유사 변환으로 구현할 수 있다.

$$v \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & p \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} e \\ f \\ q \end{pmatrix} \quad (1)$$

이는 다음과 같은 형태로 쓸 수 있다.

$$v(x, y, z) = (w(x, y), pz + q) \quad (2)$$

여기서  $w(x,y)$ 는 회전, 축소, 대칭 등을 표시하는 기하학적인 변환이고,  $pz + q$ 는 정의역 블록의 gray값에 대한을 축소변환(Contractive transformation)을 나타낸다.  $w(x,y)$ 의 회전변환과 대칭을 이용하여 평면에서 블록을 임의 각도로 회전 변환을

할수있다. 그러나, x,y가 정의역 블록의 gray값의 위치를 나타내는 좌표이므로 변환의 범위를 제한하여 0°, 90°, 180°, 270°의 회전과 각각의 대칭만을 사용한다.

## 2.4 분류화

분류화 과정은 블록의 복잡도에 따라 블록 코딩 방법을 달리해서, 블록당의 비트수를 줄여 압축율을 높이기 위한 것이다. 분류화 알고리즘은 Ramamurthi[13]의 분류기를 이용한 이웃한 화소간의 가로, 세로의 경사 정도를 가지고 평탄블럭(shade), 중간블럭(midrange), 에지블럭(edge)으로 구분하며, 평탄블럭은 블럭내의 gray값들이 평탄한 것을 말하며, 에지블럭은 블럭내의 경계가 뚜렷한 것을 말한다. 그리고, 중간블럭은 평탄블럭이 아니면서 에지블럭보다 뚜렷한 경계가 없는 블럭을 말한다.

## 2.5 프랙탈 부호화

부호화는 모든 지역 블록의 유사 변환의 계수를 찾는 과정인데, 효율적으로 블럭을 코딩하기 위해 Jacquin은 지역 블록의 복잡도에 따른 분류화를 통해 각각 다른 방법으로 코딩하였다. 즉, 블럭을 세가지로 분류하였을 때 각각의 블럭에 대한 부호화 방법은 다음과 같다.

i) 지역블럭이 평탄블럭일 때,  
블럭내의 화소들의 평균값으로 블럭을 부호화한다.

ii) 지역블럭이 중간블럭일 때,  
회전 및 대칭 변환은 제외하고 축소변환만을 사용해서 부호화하는데 축소율(contractivity)은 {0.7, 0.8, 0.9, 1.0}중에서 선택하며, 오프셋(offset)은  $\text{mean}(\text{range block}) - s * \text{mean}(\text{domain block})$ 으로 구해 지역블럭과 비슷한 블럭을 선택하여 축소율,

오프셋 및 정의역 블럭의 인덱스를 구한다. 다음 식은 한 블럭이 중간블럭으로 분류되었을 때의 부호화를 나타낸다.

$$\min(d(s * \text{domainblock}_i' [0..15] + \text{offset}), \text{range block} [0..15]) \quad (3)$$

여기서,  $\text{offset} = \text{mean}(\text{range block}) - s * \text{mean}(\text{domain block})$ 이고, i는 1에서 D까지의 범위의 수이고, D는 정의역블럭의 총갯수이다. 또 s는 {0.7, 0.8, 0.9, 1.0}의 범위를 갖는다. Domain block'[0..15]는 8 x 8크기의 정의역블럭을 2:1로 축소한 것을 나타내며, d는 유클리디언 거리를 사용해 블럭간의 비슷함 정도를 나타낸다.

iii) 지역블럭이 에지블럭일 때,

회전 및 대칭 변환과 축소 변환으로 코딩하는데 축소율은 {0.5, 0.6, 0.7, 0.8, 0.9, 1.0}중에서 선택하며, 오프셋은  $\text{mean}(\text{range block}) - s * \text{mean}(\text{domain block})$  구한다. 이렇게 해서 만든 블럭에 회전, 대칭을 사용해 지역블럭과 비슷한 블럭을 선택하여 축소율, 오프셋, 정의역블럭 번호 과 회전 및 대칭 등을 구한다.

다음 식은 한 블럭이 에지블럭으로 분류되었을 때의 부호화를 나타낸다.

$$\min(d((s * \text{domainblock}_i' [0..15] + \text{offset})_{\text{iso}}), \text{range block} [0..15]) \quad (4)$$

여기서,  $\text{offset} = \text{mean}(\text{range block}) - s * \text{mean}(\text{domain block})$ 이며, iso는 회전 변환 4개, 각각의 대칭변환으로 총 8개로 했으며, i는 1부터 D까지의 정의역블럭의 중의 하나이다. domain block'[0..15]는 8 x 8 크기의 정의역블럭을 2:1로 축소한 것을 나타내며, d는 euclidean distance를 사용해 블럭간의 비슷함 정도를 나타낸다. 이렇게 구한 계수는 정의역 블럭, 축소율, 오프셋

동 순으로 비트열로 저장된다.

## 2.6 복호화

복호화는 각 블록의 분류화에 다른 형태, 즉 평탄블럭, 중간블럭, 에지블럭으로 구분하여 행하여 진다.

i) 평탄블럭일 때,

정의역블럭의 모든 gray값들을 평균치로 대신한다.

ii) 중간블럭일 때,

영상에서 정의역블럭의 번호에 해당하는 블럭 영상을 읽어와 블럭의 gray값들을 축소한다음 오프셋을 더하여 블럭을 만든다.

iii) 에지블럭일 때,

영상에서 정의역 블럭 번호에 해당하는 블럭을 읽어와 블럭의 gray값들에 축소한다음 오프셋을 가해 회전 및 대칭을 적용하여 블럭을 만든다.

다음식은 복호화를 나타낸다.

$$((s * \text{domainblock}'_{r'}[0..15] + \text{offset}_{r'})_{\text{iso}(r)})_{R_i} \quad (5)$$

여기서,  $i$ 는 반복 횟수를 나타내며,  $r$ 은  $r$ 번째 지역블럭의 인덱스를 나타내고,  $\text{iso}[0..R]$ 은 모든 지역블럭의 회전 및 대칭 변환중의 한 변환을 나타낸다. 그리고,  $\text{domain block}'[0..15]$ 는 지역블럭 크기의 두 배인 정의역블럭을 2:1로 축소한것을 의미한다. 복호화는 전체영상에 대해 복호화를 반복할수록 원영상에 가까운 복원 영상을 얻을 수 있다.

## 3. 세그먼트 기법을 이용한 프랙탈 코드

본 논문에서 블럭을 분류하는 방법은

Jacquin의 방법과 같다. 즉 블럭은 복잡도에 따라 분류하여 부호화한다. 평탄블럭은 Jacquin의 방법과 같이 평균값으로 부호화하고, 중간 영역은 gray값의 축소율과 오프셋으로 한다. 한편 에지블럭은 세그먼트기법을 사용하여 부호화한다. 즉 에지블럭은 블럭내의 에지를 경계로하여 세그먼트로 나눈다. 각 세그먼트들은 거의 에지가 제거된 중간화된 영역이어서 중간블럭과 같은 방법을 부호화한다. 이렇게 에지블럭들을 여러 개의 중간 세그먼트로 변환할 수가 있어 종전에 프랙탈 부호화에서 오차가 가장 많았던 에지의 부호화가 없어짐으로써 보다 큰 크기의 블럭단위로 부호화할 수 있어 높은 압축을 기대할 수 있다.

### 3.1 세그먼트 분할 방법

블럭을 세그먼트로 분할할 때, 세그먼트의 갯수를 허용한 갯수이내로 하기위하여 블럭내의 임의의 gray 값을 중심좌표로 선택해서 이 둘을 중심으로 가까운 gray 값을 분리한다. 그리고, 분리된 것들의 새로운 중심 좌표는 그 gray 값들의 평균값으로 한다. 그리고 이 새로운 좌표의 중심이 새로이 분리된 그룹이 있으면, 이로 부터 새로운 중심좌표를 얻는다. 이상의 방법을 반복함에 따라 중심 좌표의 이동이 없으면 반복을 중단하고 그 그룹이 분할된 세그먼트를 얻게 된다. 여기서, 세그먼트의 최대 갯수를 2로 하였을때 세그먼트로 나뉜 결과는 Fig. 2에서 처럼 흑과 백색 부분으로 구별되며, 분할된 경계선 정보를 체인 코드로 저장하고, 복호화시 이로부터 세그먼트를 복원한다.

### 3.2 제한된 정의역블럭 구성

에지를 경계로 나누어진 세그먼트에는 에지 성분이 거의 없으므로 domain pool은 에지가 없는 블럭들로 구성하였다. domain

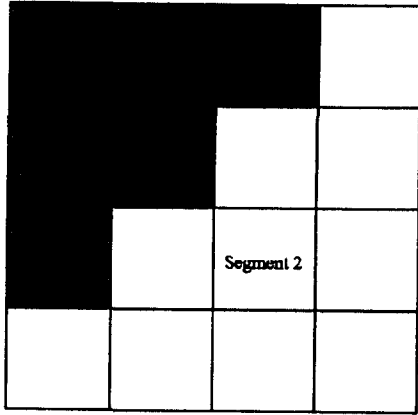


Fig. 2. Segmentation of edge block

pool은 부호화에서 축소변환되는 정의역 블록의 모임이고, 부호화에는 이의 인덱스만이 전송된다. 따라서 본 논문에서는 Hadamard 변환을 이용해서 수평, 수직 성분이 없는 블록을 택하였다.

Hadamard 변환의 가장 기본이 되는  $H_2$ 은,  $\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ 이고,  $H_3$ 는  $\begin{pmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{pmatrix}$ 이다. 그래서, 블록 사이즈가 4일 때,  $H_3$ 는 다음 행렬처럼 쓸 수 있다.

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \quad (6)$$

위 행렬을 사용해서 블록  $X$ 를  $H$ 변환했을 경우  $f_{01}, f_{02}, f_{03}$ 은 수직 벡터성분이며,  $f_{10}, f_{20}, f_{30}$ 은 수평 벡터성분이다.

$$F = HXH \quad (7)$$

$$H = \begin{pmatrix} f_{00} & f_{01} & f_{02} & f_{03} \\ f_{10} & f_{11} & f_{12} & f_{13} \\ f_{20} & f_{21} & f_{22} & f_{23} \\ f_{30} & f_{31} & f_{32} & f_{33} \end{pmatrix} \quad (8)$$

그리고,  $F_h = (f_{10}, f_{20}, f_{30})$ 과  $F_v = (f_{01}, f_{02}, f_{03})$ 을 만들어  $t_h, t_v$ 는 다음 조건으로부터 계산된다.

$$t_h = \begin{cases} \text{MIN}[ |F_h(j)| ] & , \text{ if } F_h(j) > 0 \\ -\text{MIN}[ |F_h(j)| ] & , \text{ if } F_h(j) < 0 \end{cases}$$

$$t_v = \begin{cases} \text{MIN}[ |F_v(j)| ] & , \text{ if } F_v(j) > 0 \\ -\text{MIN}[ |F_v(j)| ] & , \text{ if } F_v(j) < 0 \end{cases}$$

(9)

$$j = 1, 2, 3$$

각 블록마다  $T = \sqrt{(t_h^2 + t_v^2)}$ 이 작은 순서로 8개를 선택하여 domain pool을 결정한다.

### 3.3 Segmented fractal block coding

세그먼트화한 블록들은 에지를 경계로하여 블록을 분리했기 때문에 각 세그먼트는 중간 블록과 특성이 같다. 따라서 각 세그먼트별로 중간영역 부호화와 같은 방법으로 부호화할 수 있다. 본 논문에서는 gray 값의 축소율을 {0.7, 0.8, 0.9, 1.0}으로 제한하고 8개의 선택된 정의역블록만으로 domain pool을 구성하였다. 회전 및 대칭 등의 기하학적인 변환은 사용하지 않고 유사변환의 계수를 찾는다.

Fig. 3은 한 블록을 세그먼트로 나누고 각 세그먼트를 segmented fractal block coding하는 그림을 나타낸 것이다. 먼저 세그먼트화 과정에서 블록을 분리하고 각각의 블록마다 fractal block coding한 결과를 저장한다. 어떤 블록에서는 블록내에 뚜렷한 에지가 없으면 세그먼트로 나누지 않고 전 블록을 fractal block coding할 수도 있다.

복호화는 Jacquin의 복호화와 같이 부호화의 역으로 수행하면 된다.

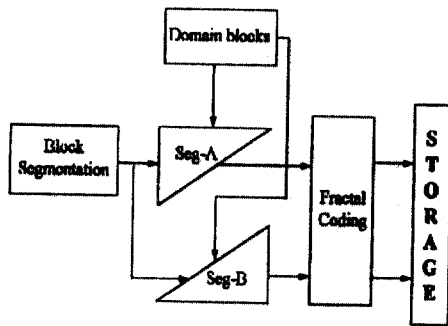


Fig. 3. The coding of segmented block

#### 4. Simulation 및 결과

시뮬레이션은 Lena 영상을 사용하였으며, 원 영상은 해상도가 128x128이며 gray level이 256이다. 이 영상을 치역블럭은 8 x 8로 하고 정의역블럭은 16 x 16으로하여 부호화하였다. 그리고, 코딩에 사용될 변수들은 한 블럭의 세그먼트 최대 갯수는 2로 하고, 유사변환의 축소율은 (0.7, 0.8, 0.9, 1.0)으로 제한하고, 8개의 정의역블럭을 선택할 수 있는 인덱스(index)번호와 오프셋으로 되어 있다. 그래서, 한 치역블럭을 부호화하는 데 필요한 비트수는 다음과 같다.

i) 평탄블럭

치역블럭의 평균 gray값으로 부호화하며  
bit 수 = 8

ii) 중간블럭

(3)식을 이용하여 계산하며

bit 수 = 2+3+8(bit)

iii) 세그먼트로 나눈 에지블럭

각 세그먼트별로 중간블럭과 같은 방법으로 부호화한다.

bit 수 = 2\*(2+3+8) + segment  
infomation(bit)

Fig 4.a)는 Lena(128 x 128)의 원 영상이고, b)는 복원된 이미지이다. 세그먼트 정보까지 추가해서 압축율은 0.42 bpp(bit per pixel)이다. 화질은 PSNR로 33dB이다. 같은 영상을 Jackin의 방법으로 압축할 때, 비슷한 PSNR을 얻는 데 압축율은 1.22 bpp[14]이므로 많은 개선이 있음을 알 수 있다.

#### 5. 결 론

낮은 전송율을 가지고 동화상을 전송하려면 높은 압축율을 가지는 알고리즘을 개발하여야 한다. 동화상의 부호화에는 정지화상에서와 같은 Intra-coding과 연속되는 이웃의 프레임과의 중복성을 이용하는 Inter-coding이 필요하다. 본 논문은 이중 Intra-coding의 압축율을 높이기 위한 방법으로 초저전송률에서의 동화상 서비스를 위한 MPEG-4에서 거론되는 방법 중에서 이론적으로 높은 압축율을 얻을 수 있는 프랙탈 부호화에 관한 연구이다. 이 방법을 사용한 영상압축은 널리 연구중이나, 아직까지는 DCT를 능가하는 결과는 미미하다. 본 논문에서 제안한 세그먼트 기법을 이용한 프랙탈 부호화는 기존의 프랙탈 부호화 방법이 에지부분의 부호화의 어려움으로 4x4의 블럭에 대하여 적용하는 것을 탈피하기 위하여 에지블럭을 에지에 따라 세그먼트로 나눈 후, 각각의 세그먼트를 중간블럭과 같이 부호화함으로써 기존에 방법보다 더 큰 블럭 단위로 부호화할 수 있도록 하였다. 시뮬레이션 결과에서 이 방법이 8x8 블럭을 이용하였을 때, 압축율이 기존의 프랙탈 방법은 물론 DCT 기반의 압축보다 높은 압축율을 얻을 수 있었다.



a) original Lena image



b) reconstructed Lena image

Fig. 4. Segmented Fractal Block coding

### 참 고 문 헌

- [1] CCITT Recommendation H.261, *Video codec for audio-visual services at  $p \times K$ bits/s*, May 1989
- [2] Committee Draft of standard /ISO/MPEG 90 / 176, *Coding of moving pictures and associated audio*. Dec. 1990
- [3] Working Draft ISO/IEC JTC1/SC29/WG11/MPEG95, *Coding of moving pictures and associated audio*. Mar. 1995
- [4] ISO/IEC JTC1/SC29/WG11, MPEG-4 *Requirements Document*, MPEG94, July 1994
- [5] M. Antonini, M. Barlaud, and et.al, "Image Coding Using Wavelet Transform," *IEEE Trans. on Image Processing*, vol. 1, no. 2, April 1992.
- [6] H. Harashima, K. Aizawa and T. Satio, "Model-Based Analysis Synthesis Coding Videotelephone Images," *Trans. IEICE Japan*, E72, 5, pp 452-459, 1989
- [7] J. Ostermann and P. Gerken, "Object-oriented analysis-synthesis coding based on source models of moving 2D and 3D-objects," ISO/IEC JTC1/SC29/WG11 MPEG93/710, July 1993
- [8] Michael F. Barnsley and Alan D. Sloan, "A Better Way to Compress Images," *BYTE*, vol. 13, no. 1, pp.215-224, January 1988.
- [9] Amaud 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, Jan. 1992.

[10] Y. Fisher, *Fractal Compression : Theory and Application to Digital Images*, Springer Verlag, New York, 1994.

[11] D. M. Monro, F. Dudbridge, "Fractal Approximation of Image Blocks," IEEE ICASSP, pp. 485-488, 1992

[12] Michael Barnsley, *Fractals Everywhere*, Academic Press, Boston 1988.

[13] Bhaskar Ramamurthi and Allen Gersho, "Classified Vector Quantization of Images," IEEE Trans. On Communications, VOL. COM-34, NO.11, Nov. 1993.

[14] 이 주엽, 반복 축소 기법을 이용한 프랙탈 영상 부호화에 대한 연구, 석사 논문, 강원대학교 대학원, 1993년 2월