

論文94-31B-2-6

블럭단위의 프랙탈 근사화를 이용한 영상코딩

(Image Coding by Block Based Fractal Approximation)

鄭賢民*, 金容奎*, 尹鐸鉉*, 姜賢哲*, 李炳來*, 朴圭泰*

(Hyun Meen Jung, Young Kyu Kim, Taek Hyun Yun, Hyun Chul Kang, Byeong Rae Lee and Kyu Tae Park)

要約

본 논문에서는 프랙탈 기법중 SAS(Self Affine System)를 사용하여 영상을 블럭단위로 근사화하는 코딩 방법을 제시한다. 각 블럭을 4개의 타일로 나누어 각각의 타일에 대해 4개의 affine mapping 계수를 구하였다. affine 변환을 적용한 영상블럭과 복원되는 영상블럭의 에러가 최소값이 되는 affine mapping의 계수를 찾기 위해, 각 계수에 대해 편미분하여 에러가 극소가 될 때의 계수값이 정해지도록 행렬식을 구하였다. 코딩되는 블럭의 계조도가 수렴하도록 균일하게 분할되는 4개의 affine 변환을 적용하였다. 자연스러운 영상복원이 가능한 프랙탈 기법의 특성을 영상코딩에 이용하기 위해 계조도의 변이가 적은 배경에 대해서는 압축효율이 높은 큰 블럭을 적용하고, 계조도의 변이가 큰 texture와 에지에 대해서는 블럭을 다시 분할하는 가변블럭 기법을 도입하였다. 블럭은 영상코딩의 효율성을 위해 16×16, 8×8, 4×4 블럭으로 구분하여 affine mapping의 각 계수를 구하였다. 또한 블럭의 계조도 변이에 따라 평탄, texture 및 에지블럭으로 구분하여 평탄블럭인 경우 평균 계조도를 전송하고, texture와 에지블럭에 대해서는 affine mapping 계수를 구하였다. 각 계수에 대해 양자화를 실시하여 한 블럭에 대해 16바이트만 전송하도록 하였다. 본 논문에서 제시한 방법을 사용하면, 영상의 크기에 대해 계산량이 선형적으로 비례하는 특성을 얻게된다. 512×512 크기의 256 계조도인 lena영상에 대해 적용한 결과 1.105bpp에서 31.58 dB를 얻었다.

Abstract

In this paper, a block based image approximation technique using the Self Affine System(SAS) from the fractal theory is suggested. Each block of an image is divided into 4 tiles and 4 affine mapping coefficients are found for each tile. To find the affine mapping coefficients that minimize the error between the affine transformed image block and the reconstructed image block, the matrix equation is solved by setting each partial differential coefficients to zero. And to ensure the convergence of coding block, 4 uniformly partitioned affine transformation is applied. Variable block size technique is employed in order to apply natural image reconstruction property of fractal image coding. Large blocks are used for encoding smooth backgrounds to yield high compression efficiency, and texture and edge blocks are divided into smaller blocks to preserve the block detail. Affine mapping coefficients are found for each block having 16×16, 8×8 or 4×4 size. Each block is classified as shade, texture or edge. Average gray level is transmitted for shade blocks, and coefficients are found for texture and edge blocks. Coefficients are quantized and only 16 bytes per block are transmitted. Using the proposed algorithm, the computational load increases linearly in proportion to image size. PSNR of 31.58dB is obtained as the result using 512×512, 8 bits per pixel Lena image.

*正會員, 延世大學校 電子工學科
(Dept. of Elec. Eng., Yonsei Univ.)

接受日字 : 1993年 3月 12日

1. 서론

영상정보에서 인간의 시각적인 특성을 이용하여 불필요한 정보를 제거하거나 주파수영역으로의 변환에 의해 정보량을 감소시키는 방법이 영상압축에서 사용되어 왔다. 정지영상의 경우, JPEG으로 표준화 되었으나 이산여현변환(Discrete Cosine Transform)을 이용하여 스펙트럼 영역에서 처리하기 때문에 압축률을 높이기 위해 AC성분을 제거함에 따라 고주파 성분인 에지가 상실되며, 영상의 확대에 따른 블러화를 유발하는 문제점으로 인해 새로운 영상압축 방법으로 프랙탈 기법을 이용하는 방법이 제기되고 있다. 프랙탈이라는 말은 라틴어의 frangere에서 유래하며 fractional dimension으로 해석할 수 있다. 70년대 후반 Mandelbrot에 의해 프랙탈 개념이 도입된 이후 프랙탈을 영상압축에 이용하는 방법은 Barnsley¹의 의하여 구체화되기 시작하였다. Barnsley는 IFS(Iterated Function System) 기법을 사용하여 영상을 압축하는 방법을 연구하였으며, 1989년 Jacquin^{2,3}은 자연영상에 대해 원래의 영상을 복원하는 정확한 affine mapping을 구하는 것이 불가능하기 때문에 영상안에서 블럭간의 자기유사성을 이용하여 영상압축을 하기 위해 중첩되는 domain블럭과 중첩되지 않는 range블럭으로 구분한 후 CVQ에서 적용된 분류기법과 IFS를 적용하는 새로운 프랙탈 블럭코딩 기법을 lena영상에 적용하였으며, Beaumont⁴는 Jacquin의 방법에서 원 영상과 평균값의 차를 이용한 개선된 방법을 제시하였고, Yuval Fisher⁵는 affine 변환에서 scaling factor와 offset을 이용하여 영상압축을 하였다. Monro와 Dudbridge⁶는 IFS(Iterated Function System)의 특수한 경우가 되는 affine 변환 시스템을 이용하여, 주어진 블럭에 대해 affine 변환을 구할수 있도록 하는, 역변환 문제에 대해 특별한 해를 제공하는 least square 근사화를 실시하였다. 즉 영상에 대해 8×8 블럭으로 나눈 후 각 블럭에 대해 collage theorem을 적용하여 복원된 영상과의 에러를 최소화시키는 값을 구하기 위해 affine mapping 근사화에서 적용되는 각 계수에 대해 편미분하여 역변환 문제를 해결하였다. 이 방법을 사용하면 영상의 크기에 계산량이 선형적으로 비례하는 장점이 있다.

본 논문에서는 원영상과 복원영상에 대한 편미분방법에 의해 affine 계수를 구할 수 있는 역변환 문제를 블럭의 크기에 무관하도록 일반화하였으며, 영상의 압축율과 화질을 높이기 위해 블럭의 계조도 특성을 구분할 수 있는 분류기를 사용하여 16×16 블럭에

대해 평탄, texture 및 에지 3가지 형태로 분류하여 평탄블럭은 계조도의 평균값을 전송하고 texture블럭은 affine mapping으로 부터 계산된 계수를 이용하여 영상압축을 실시하고, 에지는 다시 8×8 및 4×4의 크기를 갖는 부블럭으로 분할하여 압축하였다. 이러한 방법을 영상에 적용할 경우, 배경과 같이 블럭이 부드러운 texture인 경우는 프랙탈 코딩의 장점인 자연스러운 복원에 따라 큰 블럭단위로 코딩이 가능하며, 에지인 경우는 4개의 부블럭으로 세분화하여 영상을 압축한다. 이러한 알고리즘을 사용하면 세밀한 블럭에 대해 많은 정보를 할당하여 정밀한 복원이 가능해지는 장점이 있다. 복원시 원래의 영상으로 수렴해가는 반복수행 횟수에 따라 처리시간에 가변성을 부여할 수 있다.

II. 프랙탈 코딩

1. 프랙탈 코딩의 원리

프랙탈계수를 이용한 영상압축의 원리를 나타내기 위해, Sierpinski triangle의 영상이 원래의 영상보다 적은 기억공간을 이용하여 약간의 변환과정(IFS)을 거쳐 복원되는 과정을 보인다. 그림 1에서와 같은 3가지 변환은 식 (1)과 같이 나타난다.

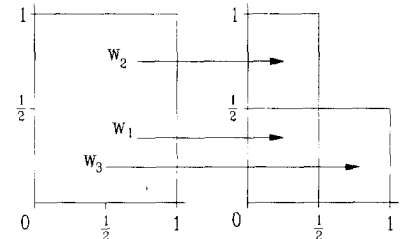


그림 1. 평면에서 3개의 affine 변환

Fig. 1. Three affine transformations in the plane.

$$w_1 \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (1-a)$$

$$w_2 \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{2} \end{bmatrix} \quad (1-b)$$

$$w_3 \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \frac{1}{2} \\ 0 \end{bmatrix} \quad (1-c)$$

식 (1)은 임의의 집합 S에 대해 다음과 같이 표현할 수 있다.

$$W(S) = \bigcup_{i=1}^3 w_i(S)$$

W를 n번 적용한 것을 W^n 으로 나타낸다.

$A_n = W(A_{n-1}) = W^n(A_0)$ 를 정의하고, A_0 를 원점 (즉, $A_0 = \{(x,y) \mid 0 \leq x \leq 1, 0 \leq y \leq 1\}$)의 왼쪽 귀퉁이로부터 단위크기를 갖는 사각형으로 선택한다. $n \rightarrow \infty$ 에 따라, 집합 A_n 은 집합 A_∞ 로 수렴한다. 실제로 어떠한 콤팩트(compact) 집합 $S \subset R^2$ 도 $n \rightarrow \infty$ 에 따라 $W^n(S) \rightarrow A_\infty$ 이 된다. 그림 2는 A_1, A_2, A_3, A_4 를 나타내며 그림 3은 A_∞ 를 나타낸다.

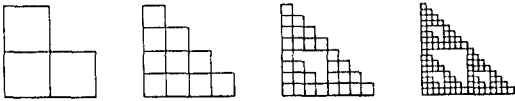


그림 2. $A_1 = W(A_0)$ 와 A_2, A_3, A_4 의 영상
Fig. 2. $A_1 = W(A_0)$ and its images $A_2, A_3,$
and A_4 .

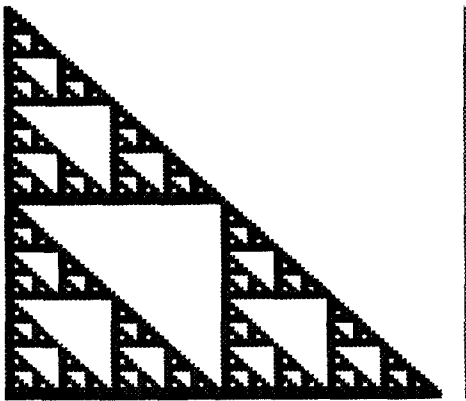


그림 3. $A_\infty = \lim_{n \rightarrow \infty} W^n(A_0)$ 의 Sierpinski 영상
Fig. 3. Sierpinski image of $A_\infty = \lim_{n \rightarrow \infty} W^n(A_0)$.

모든 콤팩트 성질을 갖는 초기집합들이 A_∞ 로 반복 수행함에 따라 수렴한다는 사실은 중요한 의미를 갖는데, 집합 A_∞ 는 단지 w_i 에 의해서 정의된다는 것을 의미한다. w_i 가 수렴하기 때문에 어떠한 초기 A_0 도 w_i 가 반복 적용됨에 따라 한점으로 축소된다.

각각의 w_i 는 6개의 실수값으로 결정되므로 총 18

개의 부동소수점이 필요하다. 즉 $18 \times 4 = 72$ 바이트의 기억용량을 필요로 하는데 Sierpinski triangle을 실제로 이진영상으로 저장하기 위해서는 $256 \times 256 \times 1 \text{ bit} = 8192$ 바이트의 기억용량이 필요하다. 이에따른 압축비는 113.8 : 1이 된다.

2. 영상코딩으로의 확장

프랙탈 영상압축에서 역변환 문제에 대해, 현재까지는 IFS가 deterministic 프랙탈에 대한 표준방식이 되고 있다. 먼저 x, y 좌표, 즉 R^2 에서 affine 변환의 집합을 고려하면 다음과 같다.

$$W = \{w_k : k = 1, \dots, N\}$$

여기서 w_k 는 다음과 같이 표현되며

$$w_k \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_{11}^{(k)} & a_{12}^{(k)} \\ a_{21}^{(k)} & a_{22}^{(k)} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1^{(k)} \\ b_2^{(k)} \end{bmatrix} \quad (2)$$

각각의 affine 변환 w_i 가 수축사상이면, 다음과 같은 유일한 attractor 집합 A 를 갖는다.

$$A = \bigcup_{k=1}^N w_k(A) \quad (3)$$

affine 변환을 실제로 영상 코딩에 이용하기 위해서는 W가 계조도에 대해 적용되도록 z축 방향으로 확장되어야 한다. 즉, 자연영상을 모델링하고 코딩하기 위해 사상 $W: F \rightarrow F$ 으로 표현한다. 여기서 F는 영상의 공간이며 사상 W는 수축을 나타낸다. 즉, W는 주어진 영상을 압축할 때 고정점으로 수렴해 나가도록 정해주어야 한다. $I = [0,1]$ 이며 I^n 은 I를 n번 cartesian곱을 적용한 집합이라고 할 때 F는 $(x,y,f(x,y)) \in I^3$ 의 공간이 된다. F에서 한점은 무한대의 해상도를 갖는 하나의 추상적인 영상으로 볼 수 있는데, $f(x,y)$ 는 영상의 한점 (x,y) 에서 계조도를 표현한다. 한정된 해상도를 갖는 영상은 격자를 이용하여 I^3 를 분할하는 것으로 모델링될 수 있다.

복원 영상이 원영상으로 수렴함을 보이기 위해 contractive mapping fixed point theorem은 complete 거리공간이 필요하기 때문에 정의된 공간과 거리가 필요하다. 여기서는 공간상의 거리가 가능한 간단히 선택되기 위해 다음과 같이 나타낸다.

$$\delta(f,g) = \sup_{x,y \in F} |f(x,y) - g(x,y)| \quad (4)$$

여기서 $f,g \in F, (F,\delta)$ 는 complete 거리공간을 생성한다.

모든 $x, y, z_1, z_2 \in \mathbb{R}$ 에 대해 다음 식 (5)를 만족하는 양의 실수 $s < 1$ 이 존재하면 사상 $w : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ 은 계조도의 수축도 s 를 갖는 z 방향 수축이라고 한다.

$$|w(x, y, z_1) - w(x, y, z_2)| < s|z_1 - z_2| \quad (5)$$

사상 W 를 식(6)과 같이 두면

$$W = \bigcup_{i=0}^n w_i \quad (6)$$

임의의 $f, g \in F$ 에 대해 식 (7)이 성립하는 양의 실수 $s < 1$ 이 존재하면

$$\delta(W(f), W(g)) < s\delta(f, g) \quad (7)$$

W 는 수축이라고 하며, s 는 W 의 수축도라 한다.

원영상 $g(x, y)$ 에 대해 역변환 문제를 바로 구현하기는 어렵다. 따라서 $g(x, y)$ 를 블럭으로 분할하여 각 블럭에 대해 IFS를 적용하면 영상의 코딩이 용이하게 된다.

collage theorem은 원영상을 정확히 복원할 수 있는 affine 변환을 구할 수 없기 때문에 가급적 원영상을 복원할 수 있는 affine 변환을 적용하기 위해 필요하며, $W: F \rightarrow F$ 가 수축도 s 를 갖는 수축이고 $f \in F$ 는 영상이라 하면

$$\delta(A, f) \leq \frac{1}{1-s} \delta(W(f), f) \quad (8)$$

로 나타나며, 평균제곱근 거리를 다음과 같이 정의하면

$$d_{ms}(f, g) = \left\{ \sum \delta(f, g)^2 / r^2 \right\}^{1/2} \quad r^2: \text{영상의 크기} \quad (9)$$

collage theorem의 변이는 다음과 같이 구할 수 있다.

$$d_{ms}(f, g) \leq (1-s)^{-1} d_{ms}(W^*g, g) \quad (10)$$

여기서 f 는 원영상 g 에 대한 불변 근사값이 된다.

영상압축에서 역변환 문제를 해결하기 위해, SAS 이론을 attractor 집합에서 정의된 함수에 적용한다. 일반적으로 affine 변환은 x, y 에 대하여 적용되지만 영상 $g(x, y)$ 에 대하여 적용하기 위해 새로운 항목 z , 즉 다음과 같이 계조도에 대하여 고려한다.

$$m_k \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_{11}^{(k)} & a_{12}^{(k)} & a_{13}^{(k)} \\ a_{21}^{(k)} & a_{22}^{(k)} & a_{23}^{(k)} \\ a_{31}^{(k)} & a_{32}^{(k)} & a_{33}^{(k)} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} b_1^{(k)} \\ b_2^{(k)} \\ b_3^{(k)} \end{bmatrix} \quad (11)$$

contractive 함수 $\{m_k(z) : A \rightarrow \mathbb{R} \rightarrow \mathbb{R} : k = 1, \dots, N\}$ 가 존재한다면, 함수 $f: A \rightarrow \mathbb{R}$ 은 불변이다. 일반적으로 x, y 방향으로는 균등한 수축도를 갖는 것이 효율적이므로 다음과 같은 식으로 표현할 수 있다.

$$m_k \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ a_1^{(k)} & a_2^{(k)} & a_3^{(k)} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} b_1^{(k)} \\ b_2^{(k)} \\ b_3^{(k)} \end{bmatrix} \quad (12)$$

식(12)를 x, y 평면과 계조도에 대한 식으로 분할하면 다음과 같다.

$$w_k \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1^{(k)} \\ b_2^{(k)} \end{bmatrix} \quad (13-a)$$

$$v_k[z] = \begin{bmatrix} a_1^{(k)} & a_2^{(k)} & a_3^{(k)} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + b_3^{(k)} \quad (13-b)$$

식(12)에 의해 한점에서 복원되는 영상계조도는 식(13)과 같이 표현할 수 있다.

$$f(p) = W^*f(p) = W_k(w_k^{-2}(p), f(w_k^{-2}(p))) \quad (14)$$

여기서 $k = \max\{i : p \in w_i(A)\}$

A 가 중첩되지 않는다면 적절한 attractor를 구성하여 어떠한 초기영상 f_0 에 대해서도 고유의 값을 복원할 수 있는 고정점, 즉 고유의 계조도를 구할 수 있다.

식(13-b)의 $a_1^{(k)}, a_2^{(k)}, a_3^{(k)}, b^{(k)}$ 에 대해 collage theorem의 우변을 최소화하기 위해 원영상 g 에 대해 프랙탈 least-square 근사화를 실시한다.

실제로 구현을 위해 식(10)을 다음과 같이 표현할 수 있다.

$$d(g, V^*g) \leq \sum_{k=1}^N c_k d(g^*w_k, V^*g^*w_k) \quad (15)$$

여기서 c_k 는 실수상수이다. $k = 1, \dots, N$ 에 대해 다음 식을 최소화 한다.

$$\int_A (g^*w_k - V^*g^*w_k)^2 = \int_{x, y \in A} [g(w_k(x, y)) - a_1^{(k)}x - a_2^{(k)}y - a_3^{(k)}g(x, y) - b^{(k)}]^2 dL \quad (16)$$

식(16)의 우변에 대해 $\frac{\partial}{\partial a_1^{i,j}}, \frac{\partial}{\partial a_2^{i,j}}, \frac{\partial}{\partial a_3^{i,j}}, \frac{\partial}{\partial b^{i,j}}$ 로 각 계수를 편미분하여 각각에 대해 0이 되도록 행렬 방정식을 구하면 식(17)과 같이 표현된다.

$$\begin{bmatrix} A & 0 & \sum_{j=T}^T (i+1/2)g(i,j) & 0 \\ 0 & A & \sum_{j=T}^T (j+1/2)g(i,j) & 0 \\ \sum_{j=T}^T (i+1/2)g(i,j) & \sum_{j=T}^T (j+1/2)g(i,j) & \sum_{j=T}^T (g(i,j))^2 & \sum_{j=T}^T g(i,j) \\ 0 & 0 & \sum_{j=T}^T g(i,j) & B \end{bmatrix} \begin{bmatrix} a_1^{i,j} \\ a_2^{i,j} \\ a_3^{i,j} \\ b^{i,j} \end{bmatrix} = \begin{bmatrix} \sum_{j=T}^T (i+1/2)g(w_i(i,j)) \\ \sum_{j=T}^T (j+1/2)g(w_i(i,j)) \\ \sum_{j=T}^T g(i,j)g(w_i(i,j)) \\ \sum_{j=T}^T g(i,j) \end{bmatrix} \quad (17)$$

여기서 A와 B는 각각 $(4/3 \cdot \text{tile}^4)$, (Blocksize^2) 이며 T는 $(\text{Blocksize} / 2)$ 이다.

III. 인코딩/디코딩 알고리즘

블럭단위로 분할된 영상을 프랙탈 근사화 방법을 이용하여 압축하기 위해서는 (x, y) 좌표의 수렴과 더불어 계조도 z값을 고정점으로 수렴시키는 방법을 구

현해야 한다. 이를위해 식 (12)를 만족하도록 인코더를 구성한다. 여기서 적용한 affine 변환은 블럭을 4개의 균일한 크기를 갖는 타일에 적용하여 사용하였다. 효율적인 압축과 복원을 위해 블럭을 평탄, texture 및 에지로 구분할 수 있는 분류기를 구현하였다.

구현된 인코딩/디코딩 알고리즘은 그림4에 나타나 있다.

1. 인코딩(Encoding)

각 블럭의 타일들에 대해 affine 변환을 적용하기 위해 적용된 w_i 는 다음과 같다.

$$\begin{aligned} w_1 \begin{bmatrix} X \\ Y \end{bmatrix} &= \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \frac{-TILE}{2} \\ \frac{TILE}{2} \end{bmatrix} \\ w_2 \begin{bmatrix} X \\ Y \end{bmatrix} &= \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \frac{TILE}{2} \\ \frac{TILE}{2} \end{bmatrix} \\ w_3 \begin{bmatrix} X \\ Y \end{bmatrix} &= \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \frac{-TILE}{2} \\ \frac{-TILE}{2} \end{bmatrix} \\ w_4 \begin{bmatrix} X \\ Y \end{bmatrix} &= \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \frac{TILE}{2} \\ \frac{-TILE}{2} \end{bmatrix} \end{aligned} \quad (18)$$

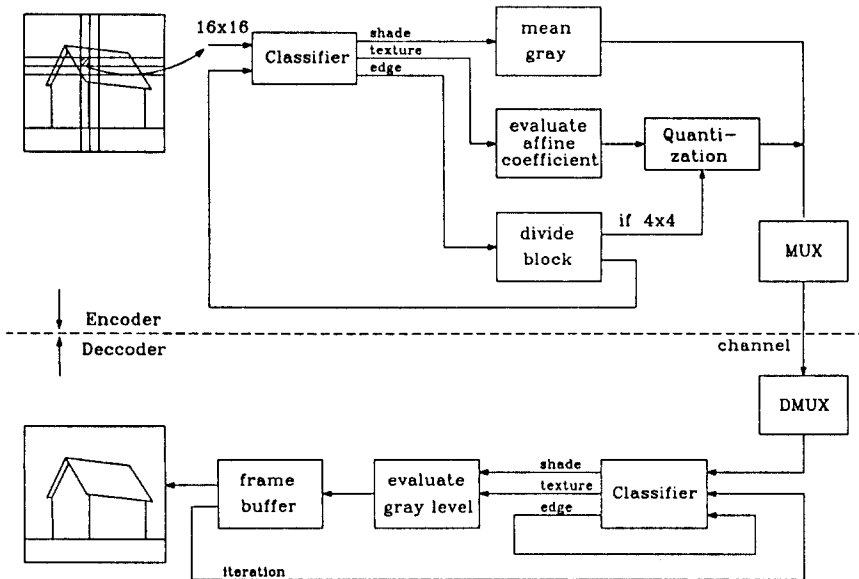


그림 4. 인코딩/디코딩 블럭도

Fig. 4. Block Diagram of Encoding/Decoding.

식 (17)에서 $g(w_k(x, y))$ 는 식 (18)의 W 로 부터 계산된다. 즉, 각 타일에 대해 W 에 의해 새로운 x , y 좌표를 계산하고 이 좌표에 위치한 계조도를 $g(w_k(x, y))$ 로 구한다. 이는 반복수행에 의해 고정점으로 수렴하는 계조도를 구하는 요인이 된다. $a_1^{(k)}$, $a_2^{(k)}$, $a_3^{(k)}$, $b^{(k)}$ 각 계수에 대해 역행렬을 계산하여 각 블럭에 대해 $4 \times 4 = 16$ 개의 정보를 만들어 낸다.

블럭의 크기가 작을수록 PSNR이 높아지며 블럭의 크기가 클수록 압축비가 높아지므로 최적으로 영상을 코딩하기 위해 다음과 같은 알고리즘을 적용한다.

- (1) 블럭크기를 16×16 , 타일크기를 8×8 로 초기화 한다.
- (2) 분류기를 통하여 평탄블럭, texture 및 에지로 구분한다.
- (3) 평탄블럭인 경우, 블럭에 대한 평균 계조도를 계산하여 전송한다.
- (4) texture인 경우, 각각의 타일방향으로 affine 변환을 적용하여 프랙탈계수 $a_1^{(k)}$, $a_2^{(k)}$, $a_3^{(k)}$, $b^{(k)}$ 를 계산한다.
- (5) 각 계수에 대해 양자화를 통해 4 : 1 압축을 실시하여 전송한다.
- (6) 에지인 경우, 8×8 블럭으로 분할하여 (2) ~ (4) 과정을 4×4 블럭이 될 때까지 반복한다.

자연영상에서 나타나는 평탄한 배경에 대해서 압축율을 높이기 위해 16×16 크기의 블럭으로 초기화를 실시하고 texture인 경우 실제로 affine 계수를 구한다. 에지인 경우, 높은 품질의 화질을 얻기 위해 다시 부분블럭으로 나누어 최종적으로 4×4 블럭이 될 때까지 반복한다.

위의 과정을 수행하여 생성된 각 계수에 대해 압축을 하기 위해 Monro는 프랙탈 계수값들이 복원될 수 있는 적절한 함수값들의 집합을 계산하여 간단히 하였다. 즉, 먼저 항상 블럭의 귀퉁이가 되는 w_k 의 고정점이 되는 α_k 를 선택하여 4개의 $f(\alpha_k)$ 를 계산하여 저장한다. 16개의 프랙탈 계수들이 16개의 함수값으로 대치될 수 있도록 $v_k(\alpha_i, f(\alpha_i)) : i \neq k$ 의 값들을 저장한다. 복원시 프랙탈 계수들은 4개의 선행방정식의 4개의 집합을 풀어서 함수값을 구함으로써 계수들을 복원하게 된다. 그러나 실험결과 정확한 원래의 값을 계산하기 불가능하기 때문에 각 계수에 대해 양자화를 실시하여 1 바이트의 크기로 압축하였다.

각 계수에 대해 1바이트로 압축한후 코딩된 영상정보를 전송할 때 블럭의 크기와 형태를 구분하기 위해 각 블럭에 대해 1~2 비트의 정보가 소요된다.

2. 디코딩(Decoding)

- (1) 전송된 정보로부터 평탄, texture 및 에지를 구분한다.
- (2) 평탄블럭인 경우 전송되어온 평균 계조도 값을 해당 블럭크기로 채운다.
- (3) texture인 경우 양자화된 계수값들을 복원하고, 블럭 크기에 따라 식(14)를 이용하여 v_k 값을 계산하여 블럭을 복원한다.
- (4) 에지인 경우, 양자화된 계수값들을 복원하고 quadtree에 따라 나뉘어지는 부분블럭에 대해 (1) ~ (3)과정을 반복한다.
- (5) 복원된 영상에 대해 수렴할 때까지 반복하여 (1) ~ (4) 과정을 반복한다.

8×8 크기의 블럭에서는 4 ~ 5회 반복수행으로 수렴하게 된다. 프랙탈 기법을 이용하여 영상에 대한 고유의 수렴계수를 구하게 되면 영상복원의 특성은 초기영상 f_0 , 즉 초기의 영상에 상관없이 반복수행이 진행됨에 따라 affine 계수에 의해 구해진 고유의 영상으로 수렴한다는 특성을 지니고 있다.

IV. 계수에 대한 양자화

256의 계조도를 갖는 영상에 대해서 affine 변환을 구현했을 때, 각 계수 a_1, a_2, a_3, b 는 4바이트 크기를 갖는 부동 소숫점 형태로 나타나며 각 타일에 대해서 계수가 존재하므로 64바이트 크기의 정보가 필요하다. 즉 16×16 블럭인 경우 4:1, 8×8 블럭에 대해서는 1:1, 4×4 블럭인 경우 1:4 압축이 되므로 효율적인 압축을 위해서 각 계수값을 1 바이트 크기를 갖도록 각 계수에 대해 양자화를 실시한다.

연립방정식에 의해 구해진 프랙탈계수는 실수형의 값으로 표현되기 때문에 저장 또는 전송을 위해서는 정수형으로 양자화 되어야 한다. 본 논문에서는 3가지 방법의 양자화기법을 제안하고 각 방법에 따른 화질과 장단점을 비교 분석한다.

1. 가상 계조도 대입 방법

블럭 내의 각 타일을 부호화하기 위해 프랙탈 계수값을 양자화하여 저장하지 않고 타일에서 4개의 고정점을 정하고 이 고정점에 대해 가상 계조도(pseudo gray level)를 설정하고 다음의 함수값을 저장한다.

$$f_i(\alpha_m) = v_i(\alpha_m, pg(\alpha_m)) \quad (19)$$

여기서 α_m : 4개의 고정점

$pg(\alpha_m)$: α_m 에서의 가상 계조도

부호기에서는 정해진 α_m 과 $pg(\alpha_m)$ 에 대해 4원 1차 연립방정식의 근으로 프랙탈 계수를 복원하면 된

다. 이 방법의 장점은 각 프랙탈 계수의 범위나 확률을 몰라도 양자화가 가능하다는 것이다.

$$0 < a_3^{(k)} < 1$$

$b^{(k)}$: 영상의 계조도 범위

2. Monro의 방법

이 방법도 프랙탈 계수값을 직접 저장하는 것이 아니고 블록에서 4개의 고정점 α_n 을 정하고 4개의 $f_k(\alpha_n)$ (α_n 은 w_k 에 의한 고정점)과 12개의 $v_k(w_k^{-1}, \alpha_n, f(w^{-1}(\alpha_n)))$, ($n \approx k$)를 저장하여 각 블록당 16개의 프랙탈 계수값을 4개의 f_k 와 12개의 v_k 함수값으로 대치하여 양자화한다. 복호기에서는 가상 계조도 대입 방법과 같이 연립방정식을 풀어 프랙탈 계수를 복원한다. 이 방법은 인접 블록 또는 타일에 큰 상관성이 존재하므로 각 f_k 또는 v_k 함수값의 차이에 대해 엔트로피 부호화가 가능하다는 장점이 있다.

V. 실험 및 고찰

프랙탈 근사화에 의한 계수산출 실험을 위하여 512×512 크기의 256 계조도를 갖는 Lena 영상에 대하여 실험을 하였다. 그림 5(a)는 코딩에 사용한 Lena 영상이며 5(b)는 복원시 원영상과 무관하게 프랙탈계수 고유의값으로 수렴함을 보이기 위해 초기영상으로 사용한 Jet 영상이다.

블록을 분류하지 않고 단순히 16×16, 8×8, 4×4 블록으로 하여 affine 변환을 적용한 후 복원시 5번 반복수행했을 때 PSNR과 압축율은 표1에 나타나 있다.

3. Lloyd-Max

이 방법으로 양자화 하기 위해서는 프랙탈 계수 $a_1^{(k)}$, $a_2^{(k)}$, $a_3^{(k)}$, $b^{(k)}$ 값의 범위와 확률 분포를 알아야 적용이 가능하다는 단점이 있으나 여러가지 영상을 통해 각 프랙탈 계수의 확률분포를 추출하여 다음식을 만족하는 양자화기를 통해 양자화 하였다.

표 1. 각 블록크기에 따른 복원된 영상의 PSNR(dB)

Table 1. PSNR of reconstructed image from each block size.

$$r_j \int_{d_j}^{d_{j+1}} p(x) dx = \int_{d_j}^{d_{j+1}} xp(x) dx \quad (20)$$

.where $p(x)$ 는 프랙탈 계수의 확률밀도 함수

	1 회	2 회	3 회	4 회	5 회	압축율
4×4 블록	20.85	32.94	36.84	37.12	37.15	1 : 1
8×8 블록	20.51	29.85	31.66	31.85	31.85	4 : 1
16×16 블록	19.08	25.59	27.25	27.43	27.45	8 : 1

여기서 프랙탈 계수의 범위는 다음과 같다.

$$-64 < a_1^{(k)}, a_2^{(k)} < 64$$



(a)



(b)

그림 5. (a) 코딩에 사용된 Lena 원영상, (b) Jet 영상

Fig. 5. (a) original Lena image used for coding. (b) Jet image.



(a)



(b)



(c)

그림 6. 5회 반복한 복원된 영상들
Fig. 6. retrieved images after 5 iterations.

그림 6은 각각의 블럭 크기로 인코딩된 영상에 대해 5회 반복수행했을 때의 영상을 나타낸다. 4×4블럭에 대한 영상(그림 6-a)에서는 원영상과 거의 차이를 보이지 않고 있으며, 8×8 블럭에 대한 영상(그림 6-b)에서는 모자의 깃부분에서 약간의 블러링 현상이 나타난다. 16×16 블럭에 대한 영상(그림 6-c)에서는 배경부분이 원영상과 매우 유사함을 알 수 있으며 에지 부분에서는 상당히 심한 블러링 현상을 나타내고 있다.

각 계수에 대해 가상계조도 값으로 복원된 영상과 Lloyd-Max 양자화에 의해 복원된 영상의 PSNR은 표 2.에 나타나 있다.

표 2. 원래의 계수와 양자화 했을 때 복원된 영상의 PSNR

Table 2. PSNR of reconstructed image by original coefficient and quantized coefficient.

		1 회	2 회	3 회	4 회	5 회
가상계조도 방법 계수당 1바이트	4×4	23.52	31.31	32.71	33.18	33.41
	8×8	24.48	28.32	28.46	28.50	28.52
	16×16	22.24	24.55	24.72	24.74	24.75
Lloyd-max 방법 계수당 1바이트	4×4	23.61	32.00	33.71	33.46	34.43
	8×8	25.45	31.26	31.62	31.62	31.64
	16×16	23.54	27.48	27.71	27.72	27.72

Monro가 제시한 양자화 방법은 4개의 방정식을 생성할 수는 있었으나, 각 계수에 대해 4개의 유일한 해를 구할 수 없었다. 가상계조도를 이용한 양자화 방법에서도 에지와 같은 특정한 부분에 대해서는 해를 찾을 수 없는 경우가 발생하여 에러를 유발하였고 이에따라 PSNR이 낮아졌다. 분류기를 사용하여 평탄, texture 및 에지를 분류한 결과는 표 3.에 나타나 있다. 분류기에서 클래스를 판단하기 위해서 블럭별로 계수를 각각 적용하였다. 표 3.에서는 16×16, 8×8 및 4×4 블럭에 대해 실험적으로 설정하였다.

표 3. 분류기를 거친 블럭의 갯수

Table 3. The number of blocks after classification.

	16×16	8×8	4×4
평탄블럭	29	24	134
texture (에지블럭)	659	1207	350
압축율	7.24 : 1		

프랙탈 영상 디코딩에서 원래의 영상 f_0 와 무관함을 보이기 위해 Jet영상에 대해 복원하였다. 1회 반복 후에 f_0 에 무관하게 영상이 복원됨을 알 수 있다.

Lena 영상에 대해 9번 까지의 반복수행에 대한 PSNR은 표 4.에 나타난다. 이에 따라 5번 반복 후에는 영상이 수렴함을 알 수 있다.

표 4. 반복수행후의 복원된 Lena영상의 PSNR
Table 4. PSNR of reconstructed Lena image after N iteration.

반복횟수	1 회	2 회	3 회	4 회	5 회	6 회	7 회	8 회	9 회
PSNR (dB)	25.47	31.17	31.56	31.57	31.58	31.58	31.58	31.58	31.58

그림 7(a)는 본 논문에서 사용한 알고리즘에 따라



(a)



(b)



(c)

그림 7. (a) 1회 (b) 2회 (c) 10회 수행후 복원된 영상

Fig 7. Reconstructed images after. (a) 1, (b) 2, (c) 10 iterations.

1회 반복수행후 복원된 Lena영상이며 7(b)는 2회 그리고 7(c)는 10회 수행하여 복원된 영상이다.

자기 유사성을 이용하는 Jacquin의 방법과 본 논문에서 제시한 방법과 처리시간을 비교하기 위해 사칙연산을 동일한 연산으로 간주하여 8×8 블럭을 부호화하는데 소요되는 계산량을 비교한 결과는 표 5에 제시되어 있다.

단 Jacquin의 uniform square partition의 경우와 본 논문에서 균일블럭에 대한 결과이다.

표 5. Jacquin 방법과 제안된 방법과의 연산량
Table 5. Comparative list of two methods.

Jacquin 방법		제안된 방법	
① range, domain 블럭의 MAE	384	① 행렬 생성	1,408
② 총 range블럭 수	32,768	② 역행렬 계산	100
③ Isometry 수	8		
④ α 의 가지수	4	총 연산수 (①+②)×4	6,032
총 연산 수 (①×②×③×④)	50,331,648		

VI. 결론

본 논문은 self affine에 의한 프랙탈 근사화로 영상을 압축/복원하는 방법을 제시하였다. Jacquin은 자기유사성에 따른 프랙탈 기법을 영상코딩에 이용하

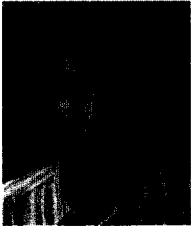
러는 영상을 처리하였다. 자기유사성에 따른 영상압축방법은 Vector Quantization과 같이 자기자신과 유사한 블럭을 찾는 데 많은 시간이 소요된다. 즉, 처리시간이 영상의 크기에 대해 비선형적으로 증가한다는 문제점을 나타내고 있다. 본 논문에서 제안한 방법은 각 블럭에 대해 직접 affine mapping 계수를 추출하는 기법을 사용함으로써 처리시간이 영상의 크기에 선형적으로 비례함을 알 수 있었다. 예지와 같이 중요한 정보에 대해서는 가변 블럭을 이용하여 정밀한 복원이 가능하며, 계조도의 변이가 거의 없는 평탄블럭에 대해서는 블럭에 대한 평균 계조도를 전송하여 코딩효율을 높일 수 있었다. 프랙탈 기법의 특징중 하나인 영상확대시 부드러운 윤곽을 나타내는 결과를 얻었으며, 3회 정도의 반복에 의해 복원된 영상의 PSNR이 거의 수렴함을 알 수 있었다. 이러한 영상의 수렴성을 이용한다면, 디코더의 성능과 해상도에 따라 횟수를 가변화 할 수 있는 가변성을 부여할 수 있다.

압축율을 고려하여 블럭을 균등한 4개의 타일로 구분하고 각 계수에 대해 1차의 편미분을 사용하여 극소값에 의해 근사화를 실시하였으므로 최적의 근사화에 대한 보장이 없다는 문제점을 내포하고 있다. 영상블럭의 효율적인 분류 방법과 최적의 affine 변환을 구하는 방법을 연구하여 동영상으로의 적용가능성에 대해 더 많은 연구가 필요하리라 생각된다.

參 考 文 獻

- [1] Michael Barnsley, Fractals Everywhere, Academic Press, 1988.
- [2] A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Trans. Image Processing* vol. 1, Jan. 1992.
- [3] A. E. Jacquin, "Fractal image coding based on a theory of iterated contractive image transformations," *SPIE* vol. 1360, Visual Communications and image Processing '90.
- [4] J. M. Beaumont, "Image data compression using fractal techniques," *BT Technol J.* vol. 9 no. 4 Oct. 1991.
- [5] Yuval Fisher, "Fractal Image Compression," SIGGRAPH '92 course notes.
- [6] D. M. Monro and F. Dudbridge, "Fractal approximation of image blocks," *IEEE ICASSP-92* vol. 3 pp.485-488, Mar. 1992.
- [7] Ramamurthi, B. and Gersho, A., "Classified vector quantization of images," *IEEE Trans. on Communications*, vol. 34, no. 11, nov. 1986.

著者紹介



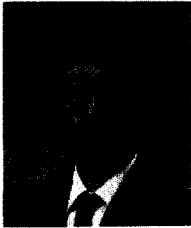
鄭賢民(正會員)

1960年 11月 5日生. 1984年 연세대학교 전자공학과 학사. 1986年 연세대학교 전자공학과 석사. 1986年 ~ 현재 한국통신 연구개발단 연구원. 1992年 ~ 현재 연세대학교 전자공학과 박사과정 재학. 주관심 분야는 영상압축, 영상처리 등임.



金容奎(正會員)

1962年 5月 5日生. 1985年 연세대학교 전자공학과 학사. 1987年 연세대학교 전자공학과 석사. 1987年 ~ 현재 삼성전자 연구원. 1991年 ~ 현재 연세대학교 전자공학과 박사과정 재학 주관심 분야는 영상압축, 영상처리 등임.



尹鐸鉉(正會員)

1963年 6月 3日生. 1986年 연세대학교 전자공학과 학사. 1993年 연세대학교 전자공학과 석사. 1985年 ~ 1991年 금성사 연구원. 1993年 ~ 현재 연세대학교 전자공학과 박사과정 재학. 주관심 분야는 영상압축, 영상처리 등임.



姜賢哲(正會員)

1959年 8月 21日生. 1982年 연세대학교 전자공학과 학사. 1984年 연세대학교 전자공학과 석사. 1989年 연세대학교 전자공학과 박사. 1990年 ~ 현재 인천대학교 공과대 정보통신공학과 조교수. 주관심 분야는 화상처리 패턴인식 등임.



李炳來(正會員)

1963年 10月 29日生. 1985年 연세대학교 전자공학과 학사. 1987年 연세대학교 전자공학과 석사. 1992年 연세대학교 전자공학과 박사. 1992年 ~ 현재 한국방송통신대학 전자계산학과 조교수. 주관심 분야는 Neural Network, 패턴인식, 영상처리 등임.

朴圭泰(正會員) 第 30卷 B編 第 10號 參照

현재 연세대학교 전자공학과 교수