DCT계수를 이용한 고속 프랙탈 압축 기법과 화상 심층암호에의 응용

이 혜 주† · 박 지 환††

요 약

프랙탈 화상압축은 원화상을 일정 크기의 블럭으로 분할하고 자기 유사성(self-similarity)을 지닌 유사 영역을 탐색한다. 이 압축방식은 부가적인 코드북의 전송이 필요하지 않아 높은 압축율을 달성하고 좋은 화질의 재생 화상을 제공한다. 이러한 이점에도 불구하고 부호화시 유사 영역의 결정을 위한 복잡한 선형변환의 수행으로 인해 부호화 시간이 증가되는 단점이 있다.

본 논문에서는 블럭의 AC(alternating current)계수들을 이용하여 선형변환의 횟수를 감소시켜 부호화 시간을 줄이는 고속 프랙탈 화상압축을 제안한다. 제안 방식은 기존의 방식과 비교하여 좋은 화질을 제공한다. 또한, 중요 기밀 데이터를 합성하는 화상 심층암호에 응용한 프랙탈 화상압축 응용법을 제시한다.

Fast Fractal Image Compression Using DCT Coefficients and Its Applications into Video Steganography

Hye Joo Lee† · Ji Hwan Park††

ABSTRACT

The fractal image compression partitions an original image into blocks of equal size and then searches a domain block having self-similarity. This method of compression achieves high compression ratio because it is unnecessary to transmit the additional codebook to receiver and it provides good quality of reconstructed images. In spite of these advantages, this method has a drawback in which encoding time increase due to a complicated linear transformation for determining a similar-domain block.

In this paper, a fast fractal image compression method is proposed by decreasing the number of transformation using AC(alternating current) coefficients of block. The proposed method also has a good quality as compared with the well-known fractal codings. Furthermore, an application of the fractal coding is proposed to apply the video steganography that can conceal an important secret data.

1. 서 론

정보통신의 발달과 함께 네트워크를 통한 멀티미디어 데이터의 전송 기회가 증가하게 되었다. 특히, 화상은 팩시밀리, 화상 회의 등 여러 분야에서 널리 이용되고 있는 가운데 기밀 데이터의 전송을 위한 화상 심층암호에도 이용되고 있다. 그러나, 화상은 데이터량이 많아 통신 비용이 증가하게 되므로 비용의 절감

※이 연구는 1995년도 한국과학재단 연구비 지원에 의한 결과임. 과제번호 951-0915-023-1

† 준 회 원:부경대학교 전자계산학과
†† 정 회 원:부경대학교 전자계산학과

논문접수:1996년 7월 3일 심사완료:1996년 10월 24일

을 위하여 화상압축이 필수적이다. 이를 위하여 지금까지 벡터 양자화, JPEG, Wavelet 등 다양한 화상압축 방식이 연구 개발되어 왔으며, 최근에는 화상 내에 자기 유사성이 존재하는 특성을 이용한 새로운 압축방식인 프랙탈 화상압축이 연구되고 있다.

프랙탈 화상압축은 1977년 Mandelbrot를 시작으로 1981년 다양성 속에 불변성을 의미하는 Hutchinson의 IFS(Iterated Function System)를 거쳐서 Barnsley의 콜라쥬 정리(Collage Theorem)를 수학적 기초로 하여 발전하였다[1]. 콜라쥬 정리는 축소 고정점 정리에 바탕을 둔 것으로 원화상 내의 자기 유사성을 발견하는데 적합한 것으로 1988년 Barnsley는 Hutchinson의 이론을 기초로 콜라쥬 정리를 이용한 최초의 프랙탈 화상압축을 제안하였다[2]. 그러나, 이 방식은 10,000:1의 높은 압축율에 비해 많은 부호화 시간이 필요할 뿐만 아니라 수동적인 방식으로 압축을 수행하였다. 따라서, 자동화된 방식으로 압축을 수행하는 새로운 방식이 Jacquin에 의해 제안되었다. 이 방식은 프랙탈 블럭 부호화로 화상을 겹침이 없는 일정 크기의 레인지 블럭(range block)으로 분할한 후, 레인지 블럭보다 크고 형태가 유사한 도메인 블럭(domain block)에 축소 변환과 회전 및 반전의 대칭변환, 그리고 화소치 변환 등으로 이루어진 선형변환을 수행하여 유사성이 높은 도메인 블럭을 선택하여 압축하는 것으로 프랙탈 화상압축의 기틀이 되었다[3, 4].

블럭의 분할 방식에 따라 프랙탈 화상압축의 압축율과 화질이 좌우된다는 인식에 의해 Fisher는 미리 정의한 최소 블럭의 크기가 될 때까지 블럭을 분할하면서 탐색하는 Quadtree 분할 방식과 화상 내의 윤곽을 따라 정방형이 아닌 블럭으로 화상을 분할하는 HV 분할을 이용한 방식을 제안하였다. 이 방식은 좋은 화질을 제공하는 반면에 탐색 시간이 많다는 단점이 있다[5].

Monro는 부호화할 블럭을 포함한 블럭이 가장 유사한 도메인 블럭이라는 가정하에 유사영역을 탐색하는 대신에 근사화 방정식을 이용하였으나, 이 방식은 블럭화가 생기는 단점이 있다. 이외에도 Oien, Lepsoy에 의해 고정된 기저 벡터를 사용하는 등의 다양한 프랙탈 화상압축이 제안되었을 뿐만 아니라, 최근에는 공간 영역에서보다는 주파수 영역에서 압축을 수행하는 방식이 제안되고 있다[6].

그러나, 프랙탈 화상압축은 하나의 레인지 블럭마다 유사영역을 찾기 위해 모든 도메인 블럭에 선형변환을 수행하여야 하기 때문에 많은 부호화 시간이 요구되는 단점으로 인해 고속화에 대한 연구가 이루어지게 되었다. 일반적으로 탐색 영역의 범위를 제한하거나 블럭의 분류 등에 의한 고속화는 재생 화질에 영향을 주게 된다. 따라서, 본 논문에서는 프랙탈 화상압축에 있어서 선형변환의 횟수를 줄임으로써 부호화 시간을 감소시키는 고속 프랙탈 화상압축을 제안함과 동시에 제안된 프랙탈 화상압축 방식을 화상 심층암호[7]에 응용한다.

화상 심층암호(video steganography)는 기밀 데이터를 몰래 전송하기 위하여 화상을 이용하는 소극적 암호의 한 방식이다. 한편, DES나 RSA와 같은 형태의 적극적 암호 방식의 경우, 공격자는 암호문에 대해 기밀 데이터의 존재를 알아 차리게 되어 암호해독의 강한 도전을 받게 된다. 그러나, 심층암호에서는 이러한 해독의 위험을 줄이기 위하여 교란용 데이터에 기밀 데이터를 합성하여 공격자가 기밀 데이터의 존재 자체를 알아차릴 수 없도록 하는 방식이다. 음성데이터를 이용하여 기밀 데이터를 합성한 이래, 1986년 Suzuki 등[8]에 의해 처음으로 교란용 데이터로써 화상을 이용하게 되었으며, 이것은 교란용 데이터가 암호문의 통계적 구조를 지배하기 위하여 데이터의 양이 많은 화상을 이용하는 것이 용이하고 화상을 정연한 구조에 잡음이 더해진 것으로 간주할 때, 인간의 눈이 잡음에는 둔감하게 반응하는 특성을 이용한 것이다.

본 논문의 구성은 2장에서 프랙탈 화상압축의 기본 개념인 축소 반복 변환에 대해서 간략하게 설명하고 제안하는 프랙탈 화상압축 방식에 대하여 기술한다. 그리고, 3장에서는 시뮬레이션을 통하여 선형변환의 횟수를 줄임에 따라 부호화 시간이 크게 감소되고, 기존의 방식과 비교하여 화질 열화가 거의 발생하지 않음을 보인다. 4장에서는 프랙탈 압축의 응용으로서 기밀 데이터의 합성법을 제시하여 양호한 화질과 충분한 크기의 기밀 데이터를 합성할 수 있음을 보인다.

2. 프랙탈 화상압축

프랙탈 화상압축의 기본 원리는 반복 축소변환 이론에 근거한다. 임의의 화상 μ, ν에 대해 μ와 ν의 차이

를 $d(\mu, \nu)$로 나타낼 때, 식(1)을 만족하는 ω를 축소변환이라 한다.

$$d(\omega(\mu), \omega(\nu)) \le s \cdot d(\mu, \nu), (s \le 1) \quad (1)$$

초기화상 μ_0에 축소변환 ω를 n회 반복적으로 수행하여 얻은 화상을 $\omega^n(\mu_0)$라 하면 다음과 같은 식이 성립됨을 알 수 있다.

$$d(\mu_{org}, \omega^n(\mu_0)) \le \frac{1}{(1-s)} \cdot d(\mu_{org}, \omega(\mu_{org})) + s^n \cdot d(\mu_{org}, \mu_0) \quad (2)$$

여기서, μ_{org}은 부호화 대상인 원화상을 의미한다. $d(\mu_{org}, \omega(\mu_{org}))$가 충분히 작은 값이고, $n \to \infty$에 대하여 식(2)의 우변은 0에 가깝게 된다. 결국 임의의 초기화상 μ_0에 축소변환 ω를 n회 반복적으로 수행한 $\omega^n(\mu_0)$는 원화상 μ_{org}에 가깝게 된다. 축소변환 ω는 일반적으로 회전, 이동 등으로 이루어진 아핀 변환(affine transform)으로 구성되어지며, ω를 나타내는 파라메터가 최종 압축 데이터로 된다.

2.1 부호화 시간

반복 축소변환은 Jacquin에 의해 처음으로 이용되었으나, 유사 블럭을 찾기 위한 탐색작업에 많은 시간이 요구되는 단점이 있다. Jacquin은 레인지와 도메인 블럭의 형태를 평탄부(shade), 중간부(mid range), 윤곽부(edge)로 나누어 블럭의 형태에 따라 정해진 축소변환 ω를 적용한다. 평탄부는 탐색에서 제외되지만, 중간부는 동일한 형태의 도메인 블럭에 미리 정해진 4개의 값을 이용한 4회의 화소값 변환(contrast scaling)을 수행한다. 그러나, 윤곽부에서의 화소값 변환은 블럭의 동적 범위(dynamic range)를 측정하여 변환치를 결정하므로 윤곽부는 1회의 화소값 변환과 8회의 대칭변환(symmetry)을 수행한다. 더우기, 다음과 같은 2단계 분할(two-level partition)방식을 이용하고 있기 때문에 변환의 횟수는 더욱 증가하게 된다. 즉, 화상을 크기가 8×8인 레인지 블럭(부모 레인지 블럭)으로 분할한 다음, 16×16 크기의 유사영역인 도메인 블럭(부모 도메인 블럭)을 탐색한다. 그리고, 유사영역의 도메인 블럭을 크기 8×8인 부블럭(자식 도메인 블럭)으로 분할하여 각 부블럭의 왜곡정도에 따라서 레인지 블럭을 4×4로 분할하여 다시 레인지 부블럭(자식 레인지 블럭)의 유사영역을 탐색하는 방식이다.

한편, 주파수 영역으로 변환하여 탐색을 수행하는 Zhao방식[6]은 레인지 블럭을 평탄부와 윤곽부로 분류한다. Jacquin방식과 달리 고정된 레인지 블럭과 도메인 블럭을 이용한다. Zhao방식에서는 비교 대상인 도메인 블럭에 8번의 화소값 변환과 8번의 대칭변환을 수행하기 때문에 더욱 많은 부호화 시간이 소요된다. 따라서, Jacquin과 Zhao방식의 1레인지 블럭에 대한 탐색시간 T는 식(3)과 같이 나타낼 수 있다.

$$\begin{aligned} Jacquin: T &= (8I + C) \times N_J(D_e) + 4C \times N_J(D_m) \\ &= 8I \times N_J(D_e) + C \times N_J(D_e) + 4C \times N_J(D_m) \\ Zhao: T &= (8I + 8C) \times N_Z(D) \\ &= 8I \times N_Z(D) + 8C \times N_Z(D) \end{aligned} \quad (3)$$

$N_J(D_e)$와 $N_J(D_m)$은 Jacquin방식에서의 도메인 블럭의 윤곽부와 중간부의 수를, $N_Z(D)$는 Zhao방식에서의 전체 도메인 블럭 수를 나타내고, I와 C는 각각 대칭변환과 화소값 변환의 수행시간을 의미한다. 식(3)에서 알 수 있듯이 선형변환의 수행에 있어서 Jacquin방식은 대칭 변환이, Zhao방식은 화소값 변환과 대칭변환이 많은 부분을 차지하고 있다. 따라서, 탐색 시간 T를 줄이기 위해서 선형변환의 수행횟수를 감소시키는 방식이 요구된다.

2.2 압축 및 복호과정

본 논문에서는 레인지와 도메인 블럭에 DCT(discrete cosine transform)를 수행하여 주파수 영역으로 변환한 후, AC계수의 부호(sign)를 이용하여 선형변환의 횟수를 줄이는 방법을 제시한다. 대칭변환에서 중복성을 배제시키고 AC계수의 부호를 이용하면 1회만의 대칭변환으로 일의성이 보장되기 때문에 부호화 시간을 대폭으로 감소시킬 수 있다.

2.2.1 압축과정

먼저 공간 영역상에서 크기가 각각 4×4, 8×8인 레인지 블럭 R_S와 도메인 블럭 D_S로 화상을 분할한다.

이와 같이 분할된 각 블럭에 DCT를 수행하여 주파수 영역으로 변환된 영역을 각각 R_F, D_F라 한다.

(1)블럭 형태의 분류

레인지 블럭과 도메인 블럭에 대해 식(4), (5)와 같이 DCT의 AC계수 절대값의 합을 임의의 임계값 T_1, T_2와 비교하여 작으면 평탄부로, 반대인 경우에는 윤곽부로 분류하여 각각 R_{Fs}, R_{Fe}, D_{Fs}, D_{Fe}로 표기한다.

$$|R_F(0,1)|+|R_F(1,0)|+|R_F(1,1)|=\begin{cases}<T_1: \text{평탄부}(R_{Fs})\\ \geq T_1: \text{윤곽부}(R_{Fe})\end{cases} \tag{4}$$

$$|D_F(0,1)|+|D_F(1,0)|+|D_F(1,1)|=\begin{cases}<T_2: \text{평탄부}(D_{Fs})\\ \geq T_2: \text{윤곽부}(D_{Fe})\end{cases} \tag{5}$$

이와 같이 분류된 레인지 블럭의 형태에 따라 유사 영역의 탐색 여부가 결정된다. 즉, 레인지 블럭이 평탄부인 경우에는 유사 영역을 탐색하지 않으며, 그 블럭의 DC계수를 나타내는 Δ_g만을 저장하고 윤곽부인 경우에는 선형변환을 수행하여 선택된 유사 영역의 위치, 파라메터 및 레인지 블럭의 Δ_g를 저장한다.

(2)유사영역의 탐색 및 선형변환

도메인 블럭에 선형변환 $\overline{D}_{Fc}=\tau[\phi(D_{Fc})]$을 수행하여 레인지 블럭과의 유사성을 측정하기 위하여 식(6)과 같은 2승오차를 이용한다.

$$d=\sum_{u=0}^{3}\sum_{v=0}^{3}(R_{Fc}(u,v)-(\overline{D}_{Fc}(u,v)))^2$$

단, $u=v=0$일 경우는 제외 (6)

변환 τ는 식(7)과 같이 대칭변환 l_n과 화소값 변환 α로 구성되며, 선형 변환을 위한 각 변환은 다음과 같다.

$$\tau[\phi(D_{Fc})]=l_n[\alpha\cdot\phi(D_{Fc})] \tag{7}$$

①축소변환:ϕ

크기가 다른 도메인 블럭을 레인지 블럭과 같은 크기로 공간 축소한다. 즉, 도메인 블럭중의 D_{Fc}를 식(8)과 같이 축소하여 D_{Fc}라 한다.

$$D_{Fc}(u,v)=\phi[D_{Fc}(\delta,\varepsilon)],\quad \begin{cases}u,v\in\{0,\cdots,3\}\\ \delta,\varepsilon\in\{0,\cdots,7\}\end{cases} \tag{8}$$

②대칭변환:l_n

프랙탈 화상압축은 〈표 1〉의 공간영역에 제시된 8개의 대칭 변환[6]을 수행하여 레인지 블럭과 도메인 블럭의 차가 최소가 되는 유사 영역을 찾게 된다. 그러나, 주파수 영역에서의 대칭변환은 〈표 1〉에서 알 수 있듯이 중복 되기 때문에 절반으로 줄일 수 있다.

〈표 1〉 주파수 영역에서의 대칭변환
〈Table 1〉 Symmetries in frequency domain

Symmetry	공 간 영 역	주파수 영역
0	(1)Identity (2)Reflection about first diagonal	$D_{Fc}(u,v)$
1	(3)Reflection about mid-vertical axis (4)Rotation through −90°	$(-1)^u D_{Fc}(u,v)$
2	(5)Reflection about mid-horizontal axis (6)Rotation through + 90°	$(-1)^v D_{Fc}(u,v)$
3	(7)Reflection about second diagonal (8)Rotation through + 180°	$(-1)^{v+u} D_{Fc}(u,v)$ $(-1)^{u+v} D_{Fc}(u,v)$

또한, 대칭변환의 결과인 $D_{Fc}'=l_n(D_{Fc})$는 단지 계수들의 부호 변화를 나타내고 있다. 이때 AC계수의 부호가 블럭내의 변환에 대한 정보를 가지는 성질을 이용하면 AC계수의 부호가 서로 다른 레인지 블럭과 도메인 블럭은 다른 형태의 블럭임을 알 수 있다. 따라서, 레인지 블럭과 도메인 블럭을 비교할 때 AC계수의 부호가 일치되도록 하여 대칭변환의 횟수를 줄일 수 있다. 여기서는 AC계수 중 지그재그 순서의 AC_{10}, AC_{01}의 부호만을 이용한다.

수행할 대칭변환을 결정하기 위해 AC_{10}, AC_{01}의 부호를 나타내는 H_1, V_1에 따라 레인지 블럭과 도메인 블럭을 〈표 2〉와 같이 그룹화 시킨다. 레인지 블럭 R_F가 그룹 Ⅰ로 분류되면 H_1, V_1은 (+, +)가 된다. 이때 유사성을 측정하는 비교 대상인 도메인 블럭 D_{Fc}의 H_1, V_1이(+, −)가 되어 그룹 Ⅱ로 판단되었을 경우를 가정한다. 이 경우, (+, −)이 (+, +)가 되기 위해서는 계수 AC_{01}의 부호를 변환시키는 대칭변환을 수행해

야 한다. 따라서, $u=0$, $v=1$의 경우이므로 〈표 1〉의 대칭변환에서 Symmetry 1인 $(-1)^v D_{Fc}(u, v)$을 수행함으로써 AC_{01}의 부호를 변환시킬 수 있게 된다. 이에 따라 레인지 블럭과 도메인 블럭의 AC_{01}의 부호를 일치시킬 수 있으며, 도메인 블럭은 레인지 블럭과 동일한 형태의 블럭으로 변환 가능하게 된다. 레인지 블럭과 도메인 블럭의 그룹에 따라 수행가능한 모든 대칭변환을 〈표 3〉에 나타낸다.

〈표 2〉 대칭 그룹
〈Table 2〉 Symmetry Group

Symmetry	H_1, V_1	그 룹
0	(+, +)	Ⅰ
1	(+, −)	Ⅱ
2	(−, +)	Ⅲ
3	(−, −)	Ⅳ

〈표 3〉 대칭변환의 수행
〈Table 3〉 Execution of Symmetries

레인지블록 / 도메인 블록	그룹 Ⅰ (+, +)	그룹 Ⅱ (+, −)	그룹 Ⅲ (−, +)	그룹 Ⅳ (−, −)
그룹 Ⅰ(+, +)	0	1	2	3
그룹 Ⅱ(+, −)	1	0	3	2
그룹 Ⅲ(−, +)	2	3	0	1
그룹 Ⅳ(−, −)	3	2	1	0

이와 같이 AC계수의 부호에 의해 블럭을 그룹화함으로서 대칭변환은 일의적으로 결정되기 때문에 1회만 수행하면 된다.

③화소값 변환: α

화소값 변환치 α는 식(9)와 같이 계산하여 {0.2, … 0.9}의 범위가 되도록 양자화 한다. 이것은 오차 d가 최소가 되는 경우로 α의 계산은 $\frac{\partial d}{\partial \alpha}=0$로부터 유도된다.

$$\alpha = Quant\left(\frac{\sum_{u=0}^{3}\sum_{v=0}^{3} D_{Fc(u,v)}' R_{Fe(u,v)}}{\sum_{u=0}^{3}\sum_{v=0}^{3} D_{Fc(u,v)}'}\right)$$

단, $u=v=0$일 경우는 제외 (9)

이와 같이 변환된 도메인 블럭 $\overline{D}_{Fc}$와 부호화 하고자 하는 레인지 블럭 R_{Fe}의 2승오차 d가 최소로 될 때의 도메인 블럭이 유사 영역으로 선택된다.

본 논문에서 제안하는 방식은 블럭을 평탄부와 윤곽부로 분류하고 윤곽부의 도메인 블럭의 수를 $N_P(D_e)$이라 할 때, 탐색 시간 T'는 식(10)과 같이 계산된다.

$$T' = (I+C)\times N_P(D_e) \quad (10)$$

이때, $N_P(D_e) \simeq N_J(D_e) + N_J(D_m)$, $N_P(D_e) \le N_Z(D)$이라 가정하면,

$$T' \simeq (I+C)\times\{N_J(D_e)+N_J(D_m)\}$$

$$= I\times N_J(D_e) + I\times N_J(D_m) + C\times\{N_J(D_e)+N_J(D_m)\}$$

$$T' \le (I+C)\times N_Z(D) \quad (11)$$

식(3)과 식(11)을 비교하면, 대칭변환의 수행 횟수를 약 1/4~1/8이상 감소시킬 수 있어 부호화 시간을 대폭 줄일 수 있다.

2.2.2. 복호화

복호는 압축 파라메터를 이용하여 임의의 초기화상으로부터 모든 레인지 블럭을 치환하면 재생화상을 얻을 수 있다. 블럭을 재생하기 위해서는 먼저 압축 데이터 중 그 레인지 블럭의 유사 영역인 도메인 블럭의 위치를 나타내는 (x, y)가 지시하는 부분을 DCT를 이용하여 주파수 영역으로 변환한 후, 축소변환 ϕ와 대칭변환 I_n을 수행한 다음 α를 이용하여 화소값을 변환 시킨다. 그리고 Δ_g의 값을 레인지 블럭의 DC계수로 하여 블럭에 역DCT를 수행하면 공간 영역상에서의 레인지 블럭으로 재생된다.

다시 전 과정을 모든 레인지 블럭에 대해 1회 이상 반복하면 결과는 일정한 화상에 수렴하게 되며, 재생화상은 원화상에 가깝게 된다. 여기서 재생시 레인지 블럭의 DC계수를 나타내는 Δ_g를 이용하기 때문에 2회의 반복으로도 근사화상에 빠르게 수렴이 이루어진다.

3. 시뮬레이션 및 평가

제안방식을 (그림 1)과 같은 256×256, 8[bit/pixel]의 표준화상 Girl과 Lenna를 대상으로 Pentium-PC (90Mhz)에서 시뮬레이션하여 재생화질, 부호화 시간 및 압축율의 관점에서 평가한다. 시뮬레이션은 Turbo C++ Ver3.0을 이용하였다.

(a)화상 Girl
(a)Image "Girl"

(b)화상 Lenna
(b)Image "Lenna"

(그림 1) 원화상
(Fig. 1) The Original Images

3.1 부호화 시간과 재생 화상의 화질

먼저, 제안방식을 평가하기 위해 선형변환시에 8개의 대칭변환을 모두 수행하는 기존 방식에 있어서 재생화상의 *SNR*변화와 부호화 시간을 〈표 4〉에 나타내었다. 이 방식에서는 임계값 T_2를 이용하지 않기 때문에 모든 도메인 블럭에 대해 비교 연산하여 레인지 블럭과 가장 유사한 도메인 블럭을 찾게 된다.

〈표 4〉 기존 방식의 *SNR*의 변화와 부호화 시간
〈Table 4〉 *SNR* and Encoding time in eight Symmetries

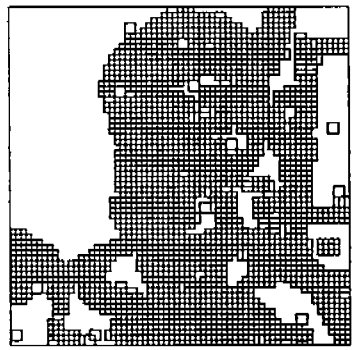
$$SNR = 20\log_{10}(255/RMS)[\mathrm{dB}],$$

$$RMS = \sqrt{\sum_{x=0}^{255}\sum_{y=0}^{255}(p(x,y)-\bar{p}(x,y))^2/255^2}, T_1 = 50$$

반복 횟수 / 화상	1	2	3	4	부호화 시간
Girl	27.034	31.493	32.049	32.057	3시간 25분
Lenna	22.700	27.813	29.287	29.321	4시간 31분

여기서, $p(x, y)$ 및 $\bar{p}(x, y)$는 원화상 및 재생화상에 있어서 (x, y)의 각 화소를 나타낸다.

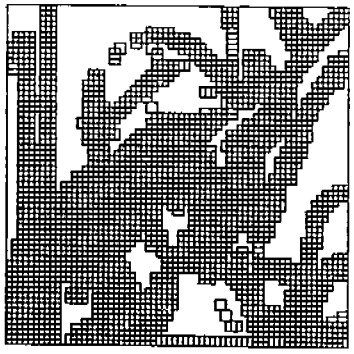
각 화상에 대해 임계값 T_1, T_2의 변화에 따른 레인지 블럭과 도메인 블럭의 분류를 〈표 5〉에 나타낸다. (그림 2, 3)은 임계값 T_2=130으로 설정하여 Girl과 Lenna의 도메인 블럭을 분류한 것으로 윤곽부가 적절히 나타나고 있음을 알 수 있다.



(그림 2) Girl의 도메인 블럭
(Fig. 2) The Domain Block of "Girl"

〈표 5〉 블럭의 분류
〈Table 5〉 Classification of Block

화상	T_1	T_2		130		100		70		50		25		0	
		레인지 블럭		도메인 블럭		도메인 블럭		도메인 블럭		도메인 블럭		도메인 블럭		도메인 블럭	
		평탄부	윤곽부	평탄부	윤곽부	평탄부	윤곽부	평탄부	윤곽부	평탄부	윤곽부	평탄부	윤곽부	평탄부	윤곽부
Girl	50	3026	1070	2173	1796	1803	2166	1385	2584	1064	2905	568	3401	0	3969
	25	2037	2059												
Lenna	50	2685	1411	1993	1976	1715	2254	1269	2700	876	3093	275	3694	0	3969
	25	1846	2250												



(그림 3) Lenna의 도메인 블럭
(Fig. 3) The Domain Block of "Lenna"

임계값 T_1이 작을 수록 많은 레인지 블럭이 윤곽부로 판단되어 유사 영역의 탐색 시간이 증가 하게 된다. 그러나, *DC*계수만으로 블럭을 재생하는 평탄부의 수가 줄어들기 때문에 재생 화상의 화질은 향상된다. 한편, 임계값 T_2가 작아 많은 도메인 블럭이 윤곽부로 판단될 경우에는 하나의 레인지 블럭에 대한 탐색 횟수가 늘어나므로 부호화 시간은 증가하지만, 부호화 화상의 성질에 따라 재생 화질에 영향을 주게 됨을 알 수 있다. 〈표 5〉와 같이 분류된 화상에 대한 부호화 시간과 4회의 반복으로 수렴한 재생 화상의 *SNR*을 〈표 6〉에 나타내었다.

$T_2=0$인 경우는 〈표 4〉의 결과인 8가지 대칭변환을 수행한 방식에 대해 선형 변환의 횟수를 감소시킨 것으로 1레인지 블럭에 대해 모든 도메인 블럭을 탐색한다. 즉, 선형변환의 횟수를 줄임으로써 부호화 시간이 대폭 줄어 들었음을 알 수 있다.

〈표 6〉 *SNR*과 부호화 시간
〈Table 6〉 The *SNR* and Encoding Time
(a) Girl 화상의 *SNR*과 부호화 시간
(a) The *SNR* and Encoding Time of "Girl"

시간 : [Sec.]

T_1 \ T_2		200	130	100	70	50	25	0
50	*SNR*	32.109	32.057	32.015	31.947	31.927	31.916	31.916
	부호화 시간	78	139	179	233	270	338	428
25	*SNR*	33.571	34.015	34.108	34.109	34.072	34.005	34.001
	부호화 시간	147	264	341	444	515	644	817

(b) Lenna 화상의 *SNR*과 부호화 시간
(b) The *SNR* and Encoding Time of "Lenna"

시간 : [Sec.]

T_1 \ T_2		200	130	100	70	50	25	0
50	*SNR*	28.962	29.035	29.029	29.004	28.987	28.974	28.877
	부호화 시간	159	231	270	327	377	474	535
25	*SNR*	29.611	29.868	29.918	29.943	29.946	29.899	29.889
	부호화 시간	252	366	428	513	598	751	849

Girl은 변화가 적고 반대로 Lenna는 변화가 많은 화상으로 성질이 다르다. 〈표 6(a)〉의 결과를 보면 T_1이 작아짐에 따라 다수의 레인지 블럭이 윤곽부로 분류

되기 때문에 화질이 높아지는 반면, 작은 T_2의 값에 대하여 오히려 SNR이 떨어짐을 알 수 있다. 이것은 T_2가 작으면 변화가 적은 도메인 블럭도 윤곽부로 분류되어 적절하지 않은 도메인 블럭이 유사 영역으로 결정되어 버릴 가능성이 있기 때문이다. 그러나, T_2가 너무 커져도 비교되는 도메인 블럭이 적어져 화질은 다시 떨어지게 된다.

반대로 Lenna와 같이 변화가 많은 화상은 임계값 $T_1=25$인 경우에는 윤곽부의 레인지 블럭이 많고, T_2가 작아 질수록 윤곽부 형태의 도메인 블럭이 많아지기 때문에 SNR이 높아진다. 그러나, 임계값 $T_1=50$인 경우에는 레인지 블럭의 수가 〈표 5〉에서와 같이 50%이상이 평탄부로 판단되기 때문에 Girl과 마찬가지로 T_2가 작아질 수록 화질은 떨어진다. 따라서, 〈표 5, 6〉으로부터 변화가 적은 화상에 대해서는 윤곽부의 형태를 갖는 레인지 블럭의 수가 많도록 T_1과 T_2를 설정하고, 부호화 시간을 줄이기 위해서 T_2는 크게 설정하는 것이 효과적이다. (그림 4)와 (그림 5)는 각 화상에 대해 가장 양호한 SNR을 얻은 문턱치일 때의 재생 화질을 나타낸 것으로 시각적으로 원화상과 거의 구별이 되지 않음을 알 수 있다.

(그림 4) Girl의 재생화상
(Fig. 4) The Decoded Image of "Girl"($T_1=25, T_2=70$)

(그림 5) Lenna의 재생화상
(Fig. 5) The Decoded Image of "Lenna"($T_1=25, T_2=50$)

3.2 압축율

화상의 재생을 위한 압축 파라메터는 블럭의 형태에 따라 〈표 7〉과 같이 저장된다. 여기서, I_c는 평탄부와 윤곽부 블럭을 판단하는 비트로써 모든 레인지 블럭에 공통적으로 부가되는 파라메터이다.

〈Table 7〉 The Number of Bits for Compression Parameter
〈Table 8〉 Compression Ratio[bit per pixel]

블럭형태	파라메터	비트수
평탄부	Δ_g	10
윤곽부	도메인 블럭의 위치	6+6=12
	Δ_g	10
	α	3
	I_n	2
	I_c	1

제안 방식에서는 임계값 T_1에 따라 압축율을 정할 수 있다. 복잡한 화상인 경우에는 만족할 만한 화질을 얻기 위해서는 T_1의 값을 작게 설정하여야 하지만, 이 경우에는 다수의 레인지 블럭이 윤곽부로 분류되기 때문에 압축율이 저하되는 상관관계가 있다. 임계값 T_1의 변화에 따른 압축율은 〈표 8〉과 같으며, 제안 방식과 비교하기 위해 문헌[9]에 제시된 압축율도 함께 나타 낸다.

〈표 8〉의 결과, 제안방식의 압축율은 $T_1=50$인 경우에 0.14, 0.05[bpp]정도 떨어지나, 더욱 양호한 화질을 얻을 수 있으며, 문헌[9]의 방식은 화소값 변환이 수행되지 않기 때문에 실제로는 거의 비슷한 압축율을 보이게 된다.

블럭 분류의 관점에서 제안 방식과 유사하게 DCT 계수를 사용하는 문헌[9]의 방식은 변환 과정을 공간 영역에서 수행하나, 제안 방식은 전 과정을 주파수

〈표 8〉 압축율[bit per pixel]
〈Table 8〉 Compression Ratio[bit per pixel]

화 상	제안 방식			문헌[9]	
	T_1, T_2	압축율	SNR	압축율	SNR
Girl	50, 130	0.965	32.057	0.8291	31.14
	25, 70	1.221	34.109		
Lenna	50, 130	1.053	29.035	0.9962	27.43
	25, 50	1.271	29.946		

영역에서 수행하기 때문에 1회의 대칭변환만을 수행한다. 따라서, 문헌[9]의 방식에 비하여 제안 방식이 빠른 부호화 시간을 달성하게 된다. 또한 Jacquin, Zhao, 문헌[9]의 부호화 시간을 식(3)과 식(11)의 결과와 비교하면 제안 방식의 부호화 시간이 향상됨을 알 수 있다.

4. 화상 심층암호에 응용

지금까지 기술한 프랙탈 화상압축 과정에 기밀 데이터를 몰래 집어넣어 전송하는 화상 심층 암호에 응용이 가능하다. 화상 심층암호는 화상 내에 기밀 데이터를 잡음의 형태로 합성하여 제3자가 기밀 통신 자체를 인식하지 못하도록 하는 방식으로 오차를 이용한 방법 등 다양한 방식이 제안되어 있다[7].

여기에서는 길이 L비트인 기밀 데이터 $M=\{m_k|k=1, \cdots, L\}$, $m_k \in \{0, 1\}$에 대해서 프랙탈 화상압축을 수행하면서 압축 파라메터 중 블럭의 대표값을 나타내는 Δ_g를 이용하여 기밀 데이터를 합성하는 두 가지의 방법을 제안한다.

4.1 방법 I

화상 내에 기밀 데이터를 집어넣는 방법으로 오차를 이용하는 방식 중에서 적응형 이산 코사인 부호화(ADCT:Adaptive DCT)를 이용하는 방법이 있다. 이 방식은 DCT를 수행하는 과정에서 기밀 데이터 m_k가 비트 '0'인지 '1'인가에 따라 DC계수를 가장 근처의 짝수/홀수로 양자화하는 것으로 본 논문에서는 레인지 블럭의 DC계수인 파라메터 Δ_g를 짝수화/홀수화한다. 그리고, 공격에 대한 안전성을 높이기 위하여 키 $K_l(l=1, \cdots, L)$을 이용한다. 키 K_l은 스트림 암호 중에서 BRM(Binary Rated Multiplexer) 시스템을 이용하여 생성하고, 아래와 같은 알고리즘에 따라 기밀 데이터를 합성한다[10].

```
Procedure Embedding_1(K_l, m_k, Δ_g)
begin
  if K_l = 1 then        ; K_l = 1이고 m_k = 1이면 Δ_g를 짝수화
    if m_k = 1 then
      if Δ_g mod 2 = 0   then Δ_g' = Δ_g
                         else Δ_g' = Δ_g + 1
    else
      if Δ_g mod 2 ≠ 0   then Δ_g' = Δ_g
                         else Δ_g' = Δ_g - 1
  else
    if m_k = 1 then      ; K_l = 0이고 m_k = 1이면 Δ_g를 홀수화
      if Δ_g mod 2 = 0   then Δ_g' = Δ_g - 1
                         else Δ_g' = Δ_g
    else
      if Δ_g mod 2 ≠ 0   then Δ_g' = Δ_g - 1
                         else Δ_g' = Δ_g
end
```

이 방식에서는 레인지 블럭당 1비트의 기밀 데이터가 합성되고, 모든 레인지 블럭에 위의 합성 알고리즘을 수행한 후 압축 데이터를 수신측에 전송한다. 수신측에서는 아래의 추출 알고리즘에 의해 Δ_g'로 부터 기밀 데이터 M을 추출할 수 있다.

```
Procedure Extract_1(K_l, Δ_g')
begin
  if K_l = 1 then
    if Δ_g' mod 2 = 0    then m_k = 1
                         else m_k = 0
  else
    if Δ_g' mod 2 ≠ 0    then m_k = 1
                         else m_k = 0
end
```

4.2 방법 II

두번째 방법으로 프랙탈 화상압축을 수행하면서 1블럭당 1비트 이상의 기밀 데이터를 집어넣기 위한 합성법을 보인다. 이 방식은 블럭의 평균을 나타내는

Δ_g의 크기에 따라 일정 비율의 하위 비트를 기밀 데이터로 대체한다.

Procedure Embedding_2(Δ_g, M_i)
begin
  if $0 \le \Delta_g < 64$ then $\Delta_g \leftarrow \{(\Delta_g$ AND 0xfffe) OR $m_{i,1}\}$
  if $64 \le \Delta_g < 128$ then $\Delta_g \leftarrow \{(\Delta_g$ AND 0xfffc) OR $m_{i,2}\}$
  if $128 \le \Delta_g < 256$ then $\Delta_g \leftarrow \{(\Delta_g$ AND 0xfff8) OR $m_{i,3}\}$
  if $256 \le \Delta_g$ then $\Delta_g \leftarrow \{(\Delta_g$ AND 0xfff0) OR $m_{i,4}\}$
end

여기서, M_i는 기밀 데이터 M의 i번째 비트의 위치를 나타내며, $m_{i,j}$는 기밀 데이터 중 i번째 비트로부터 j개의 비트를 취하는 것을 의미한다. 이와 같이 Δ_g의 크기에 따라 합성되는 기밀 데이터의 비트 수를 결정함으로써 1블럭당 1비트 이상을 합성할 수 있으며, 수신측에서는 Δ_g의 값에 따라 원래의 기밀 데이터를 추출할 수 있다. 따라서, 방법Ⅱ에서는 화상에 있어서 Δ_g의 분포에 따라 합성 가능한 기밀 데이터의 길이가 달라지게 된다.

〈표 9〉 기밀 데이터 합성시의 *SNR*의 변화
〈Table 9〉 The Change of the *SNR* at Embedding Secret Data

(a) 방법Ⅰ을 이용한 기밀 데이터 합성
(a) Embedding Secret Data With Method Ⅰ

화상	T_1 \ T_2	200	130	100	70	50	합성 비트수
Girl	50	32.104	32.050	32.007	31.936	31.918	4,096
	25	33.564	34.002	34.096	34.095	34.059	
Lenna	50	28.965	29.035	29.028	29.007	28.990	4,096
	25	29.607	29.874	29.919	29.946	29.948	

(b) 방법Ⅱ를 이용한 기밀 데이터 합성
(b) Embedding Secret Data With Method Ⅱ

화상	T_1 \ T_2	200	130	100	70	50	합성 비트수
Girl	50	31.926	31.861	31.813	31.743	31.720	13,688
	25	33.294	33.696	33.761	33.760	33.700	
Lenna	50	28.852	28.898	28.889	28.867	28.848	14,808
	25	29.470	29.704	29.739	29.762	29.762	

4.3 방법Ⅰ과 방법Ⅱ의 비교

각각의 방법으로 기밀 데이터를 합성한 후 4회 반복하여 복호하였을 때의 *SNR*의 변화를 〈표 9〉에 나타내었다.

〈표 6〉과 비교하여 방법Ⅰ은 약 0.01dB이내로, 방법Ⅱ는 약 0.4dB이내의 약간의 열화가 발생하지만 시각적으로 식별할 수 없을 정도의 양호한 화질을 얻었다. 또한, 블럭당 1비트의 기밀 데이터가 합성되는 방법Ⅰ이 방법Ⅱ보다 양호한 재생화질을 보이고 있다. 그러나, 집어넣을 수 있는 기밀 데이터의 비트수에서 방법Ⅰ은 4,096 비트로 고정되는 반면에 방법Ⅱ는 응용에 따라 가변적으로 합성할 수 있는 이점이 있다. (그림 6, 7)에 각 방법에서의 재생 화상을 나타내었다.

(a) 방법Ⅰ의 재생 화상
(a) The Decoded Image of Method Ⅰ

(b) 방법Ⅱ의 재생 화상
(b) The decoded Image of Method Ⅱ

(그림 6) 기밀 데이터 합성시($T_1 = 25, T_2 = 130$)
(Fig. 6) Embedding Secret Data($T_1 = 25, T_2 = 130$)

(a)방법 I의 재생 화상
(a) The Decoded Image of Method I

(b)방법 II의 재생화상
(b) The Decoded Image of Method II

(그림 7) 기밀 데이터 합성시($T_1=25, T_2=25$)
(Fig. 7) Embedding Secret Data($T_1=25, T_2=25$)

5. 결 론

프랙탈 화상압축은 평탄부와 윤곽부 블럭의 분류에 따라 압축율과 재생 화상의 화질이 많은 영향을 받는다. 레인지 블럭 중에서 윤곽부의 수가 많으면 좋은 화질의 재생 화상을 얻게 되지만, 선형변환에 대한 파라메터의 저장으로 인하여 압축율이 떨어지고 많은 유사 영역의 탐색으로 부호화 시간은 증가된다.

따라서, 본 논문에서는 유사 영역의 탐색시에 수행되는 선형변환 중에서 대칭변환의 횟수를 *AC*계수의 부호를 이용하여 1회로 감소시켜 부호화 시간을 줄이는 방식을 제안하였다. 제안 방식은 8회의 대칭변환을 수행한 경우와 비교하여 대폭적으로 부호화 시간을 줄이면서 재생 화상의 화질은 거의 떨어지지 않음을 알 수 있었다. 나아가, 프랙탈 화상압축 과정에 기밀 데이터를 몰래 집어넣는 화상 심층암호에의 응용을 제시하였다.

향후의 과제는 블럭의 형태를 분류하기 위한 문턱치의 설정을 적응적으로 할 수 있는 기준에 관한 연구가 수행되어야 할 것이다.

참 고 문 헌

[1] M.F.Barnsley, "Fractals Everywhere", 2nd Ed., Academic Press, 1993.

[2] M.F.Barnsley and A.D.Sloan, "A Better Way to Compress Images", BYTE, pp.213-223, 1988.

[3] A.E.Jacquin, "Image Coding Based on a Fractal Theory of Iterated Contractive Image Transforms", IEEE Trans. on Image Processing, Vol.1, No.1, pp.18-30, 1992.

[4] A.E.Jacquin, "Fractal Image Coding: Review", Proc. of The IEEE, Vol.81, No.10, pp.1451-1465, 1993.

[5] Y.Fisher, "Fractal Image Compression-Theory and Application", Springer-Verlag, 1995.

[6] Y.Zhao and B.Yuan, "Image Compression Using Fractals and Discrete Cosine Transform", Electronics Letters, Vol.30, No.6, pp.474-475, 1994.

[7] 松井甲子雄, "畫像深層暗號", 森北出版(株), p.185, 1993(in Japanese)

[8] H.Suzuki et al, "算術符號 を利用した畫像深層暗號", 1986 Symposium on Cryptography and Information Security(SCIS)(in Japanese)

[9] 박경배 외, "프랙탈 영상 부호화용 블럭 분류기", 한국정보처리학회 논문지, 제2권 제5호, pp.691-700, 1995.

[10] 이혜주, 박지환, "프랙탈 화상압축상에 기밀 데이터를 합성하는 방법", 한국정보처리학회 '96춘계 학술논문 발표집, 제3권 제1호, pp.391-396, 1996.

이 혜 주

1994년 부산수산대학교 전자계산학과 졸업(학사)
1996년~현재 부경대학교 대학원 전자계산학과 재학중(석사)
관심분야: 멀티미디어 압축, 화상처리, 암호학 응용 등

박 지 환

1984년 경희대학교 전자공학과 (공학사)
1987년 일본 국립전기통신대학 대학원 정보공학과(공학석사)
1990년 일본 요꼬하마국립대학 대학원 전자정보공학과 (공학박사)
1990년~1996년 7월 부산수산대학교 전자계산학과 전강, 조교수, 부교수
1994년~1995년 동경대학 생산기술연구소 객원 연구원
1996년~현재 동경대학 생산기술연구소 협력 연구원
1996년 7월~현재 부경대학교 전자계산학과 부교수
관심분야: 멀티미디어 압축, 암호학 응용, 오류 제어 부호, 화상처리 등