

물체 기반 프랙탈 영상 부호화

정희원 강현수*, 김성대**, 윤병주**, 조순제***

Object Based Fractal Image Coding

Hyun-Soo Kang*, Seong-Dae Kim** Regular Member, Byung-Ju Yun**,
Soon-Jae Cho*** Regular Members

요 약

이 논문은 보통의 영상에 대한 표준 프랙탈 부호화 기법과 유사하게 PIFS (partitioned iterative function system) 에 기초한 물체 기반 프랙탈 부호화 기법에 대해 제안한다. 제안된 물체 기반 프랙탈 부호화 기법에 있어서 분할된 블록들은 그들의 모양에 따라 경계 블록과 내부 블록으로 나누어진다. 경계 블록은 물체의 경계에 위치하고 임의의 모양을 가지는 블록인 반면, 내부 블록은 물체의 내부에 위치하고 사각형 모양을 가지는 블록이다. 내부 블록들은 정의역 풀(domain pool)이 영상이 아니라 물체라는 점을 제외하고는 보통의 프랙탈 부호화 기법의 과정에서 나타나는 블록들과 차이가 없으므로 내부 블록들에 대한 부호화는 기존의 프랙탈 부호화 방식과 같은 방식에 의해 부호화된다. 그러나 내부 블록들과는 달리 경계 블록은 모양이 임의의 모양을 가지고 그것의 정의역 블록이 내부 블록들로 구성되는 특징을 가지므로 다른 부호화 방식이 필요하다. 경계 블록의 이러한 특징으로부터 정의역 블록과 치역 블록의 모양의 변형에 의한 3가지 부호화 방식이 제안되고 정의역 블록이 물체 내부에 존재한다는 특성을 이용하여 내부 블록을 우선적으로 부호화하고 복원한 뒤 경계 블록의 부호화를 수행하는 방법을 제시한다. 제안된 방식들은 수학적으로 분석되고 실험을 통해 성능을 비교하고 분석 결과를 검증한다.

ABSTRACT

This paper presents an object based fractal coding scheme based on PIFS (partitioned iterative function system) like the standard fractal coding schemes for images. The partitioned blocks in our object based fractal coding scheme are classified to boundary blocks and internal blocks according to their shapes. The boundary blocks are located in the boundary region of an object and their shapes are arbitrary one, whereas the internal blocks are located in the internal region of an object and their shapes are square one. As the internal blocks make no difference with the blocks to appear in the process of the ordinary fractal coding except that domain pool is not an image but an object, encoding for the internal blocks follows the ordinary fractal coding schemes as they are. However, the boundary blocks should be carefully handled, as they have properties to be different from the internal blocks, i.e., they have arbitrary shapes and their domain pool is composed of the internal blocks not the boundary blocks. Considering these properties, we propose 3 methods to encode the boundary blocks by making an alteration to the shapes of range blocks and domain blocks, and a method to do them by using the pre-decoded internal blocks. The methods are mathematically analyzed and their performance is experimentally evaluated.

* 현대전자 정보통신(원) 기반기술연구소(khs@hei.co.kr)

** 한국과학기술원 전기및전자공학과 (sdkim, bjisyun @ sdvision.kaist.ac.kr)

*** 삼성전자 (csjss@samsung.co.kr)

논문번호 : 98537-1216 , 접수일자 : 1998년 12월 16일

I. 서론

프랙탈 영상 부호화의 핵심은 영상을 부호화하는 대신 그 영상을 발생시킬 수 있는 변환을 부호화함으로써 압축 효과를 얻는 것이다. 이것은 Barnsley^[1]에 의해 소개된 콜라주 정리(Collage theorem)와 IFS(iterative fuction system)의 수학에 기초한다. Jacquin^[2]이 처음으로 영상에 대한 실용적인 프랙탈 부호화 방식을 소개한 이래, 많은 연구들이 그의 방식을 개선하였다. 프랙탈 부호화는 지역 블록과 닮은 정의역 블록을 찾는 범위에 의해 전역방식(global method)과 지역방식(local method)으로 크게 두개의 부류로 나누어 볼 수 있다. 전역 방식은 넓은 범위에 대해 정의역 블록을 찾는 방식으로서 많은 정합 과정을 수행하게 된다. Jacquin의 방식이 전역방식의 하나이다. 이 방식이 많은 계산량을 요구하는 단점을 해결하기 위해 복잡도에 따라 블록을 분류하는 방식^{[3][4]}, 계층적으로 찾는 방식^{[5][6]}, 주파수 영역에서 찾는 방식^[7], 등이 제안되었다. 또한 웨이브렛 영역에서의 프랙탈 부호화는 블록의 복잡도에 따라 자동적인 분류가 수행됨으로써 계산량을 줄이는데 기여하였다^[8]. 한편, 지역방식은 정의역 블록을 찾는 영역을 지역 블록 주위의 몇 개의 블록으로 한정하는 방법이다. 대표적인 방법으로는 Monro의 방식^[9]과 3차원 블록에 적용한 Li의 방식^[10], 등이 있다. 이 방식들은 정의역 블록을 찾는 영역이 좁은 영역으로 제한되기 때문에 계산량에 있어서 전역방식에 비해 상당히 작은 반면 정합에 의해 발생하는 오차가 크므로 이를 다항식 근사화 하는 항을 변환에 추가하여 부호화되는 영상의 화질을 보강해 주도록 하였다. 여기서 정합은 지역 블록과 변환된 정의역 블록 사이의 정합을 의미한다.

복화기에 있어서도 성능 개선을 위한 노력이 있었다. Leopsy^[11]는 반복 변환이 필요 없는 방식을 제안하였고, Kang^[12]은 공간 변위항의 분포와 변환된 블록의 결과를 이용함으로써 빠른 수렴 속도를 얻도록 하였다.

프랙탈 부호화의 성능은 원영상과 변환의 끝개 사이의 거리에 의해 결정되므로 그 거리를 최소화 하는 변환을 찾는 것이 중요하다. 우리는 그 거리를 최소로 하는 변환을 최적의 변환이라고 정의할 수 있다. 그러나 최적 변환을 찾는 것은 너무 많은 계산량을 요구하므로 실제적으로 구현하는 것은 불가

능하다. 그래서 일반적인 프랙탈 부호화기는 콜라주 정리를 이용하여 원영상과 콜라주 영상과의 거리를 최소화 함으로써 원영상과 끝개 사이의 거리를 제한하는 방식을 사용한다.

지금까지 언급된 방법들은 앞으로 보통의, 표준의, 기존의 프랙탈 부호화 기법이라고 불려진다. 표준 프랙탈 부호화 기법들은 임의의 모양을 가지는 물체 기반 영상 부호화에 그대로 적용될 수 있다. 그러나 물체는 임의의 모양이라는 점이 고려되어야 한다. 그래서 우리는 물체의 윤곽선 정보는 칼라 정보와는 독립적으로 적당한 부호화 기법에 의해 부호화된다고 가정한다. 물체의 경계 정보가 주어짐에 따라 칼라 정보는 기존의 프랙탈 부호화 방식에 의해 부호화 될 수 있다. 즉, MPEG4와 유사하게 물체는 겹치지 않는 블록들로 나누어지고 그 블록들은 표준 프랙탈 부호화 하는 방식에 의해 부호화 된다. 이렇게 나뉘어진 블록들은 그 모양과 블록의 위치에 따라 경계 블록과 내부 블록으로 나누어진다. 여기서 내부 블록들은 사각 블록인 반면 경계 블록은 사각 블록이 아니다. 제안된 물체 기반 프랙탈 부호화 기법은 크게 내부 블록에 대한 부호화와 경계 블록에 대한 부호화로 구성되어 있다. 표준 프랙탈 부호화 기법은 내부 블록들에 대한 부호화에 거의 수정없이 적용될 수 있다. 그래서 이 논문에서는 아래에 열거된 경계 블록의 특징들을 고려하면서 임의의 모양을 가진 경계 블록들에 대한 부호화 기법에 대해 자세히 논의한다.

- 경계 블록은 임의의 모양을 가지고 물체의 경계에 위치한다.
- 경계 블록의 정의역 블록의 모양은 경계 블록의 모양과 같을 필요는 없다.
- 윤곽선 정보가 주어지는 경우 경계 블록의 모양을 변형해도 복원이 가능하므로 경계 블록의 모양은 부호화하기 편하도록 또는 부호화 성능 향상을 위해 변형될 수 있다.
- 경계 블록의 정의역 블록은 일반적으로 내부 블록들만으로 구성된다.

이 논문은 위와 같은 점들을 고려하면서 경계 블록들에 대한 프랙탈 부호화 방법들을 제안하고 분석한다. 즉, 지역 블록과 정의역 블록의 모양을 변형한 프랙탈 부호화하는 방식들과 경계 블록의 정의역 블록이 물체 내부인 점에 착안하여 성능을 개선하는 방식이 제안되고 분석된다. 제4장에서 제안

된 경계 블록에 대한 부호화 방식들을 자세히 설명하고 5장에서 실험적으로 그들을 평가한다.

II. 이론적 배경

디지털 영상의 거리 공간을 (X, d) 이라고 하자. 여기서 $d(\cdot, \cdot)$ 는 거리를 의미한다. 그때 다음 식을 만족하는 변환 $w: X \rightarrow X$ 를 생각해 보자. 디지털 영상의 거리 공간을 (X, d) 이라고 하자. 여기서

$$d(w(x), w(y)) \leq d(x, y), \quad x, y \in X \quad (1)$$

여기서 $|w| < 1$ 이면, w 는 축소 계수가 인 축소변환이라고 한다. w 가 축소 변환이면, 그 변환은 다음 식과 같은 고정점(또는 끌개) x_f 를 가진다.

$$\lim_{n \rightarrow \infty} w^n(x_0) = x_f \quad (2)$$

$$w(x_f) = x_f \quad (3)$$

식(2)는 프랙탈 부호화의 원리를 잘 보여준다. 즉, 변환 w 의 부호화는 x_f 의 부호화와 같다는 것을 의미한다. 원영상 x 와 닮은 끌개를 생성할 수 있는 변환을 원영상 대신 부호화함으로써 압축 효과를 얻는 것이다. 전체 영상에 대한 하나의 변환을 찾는 것은 힘들기 때문에 영상에 대한 변환은 전체 영상을 블록으로 나누고 나누어진 블록 변환들의 합으로 얻어진다. i 번째 블록에 대한 블록 변환을 w_i 라고 할 때 전체 영상에 대한 변환 w 는 다음 식과 같이 표현된다.

$$w(x) = \cup_{i=0}^{N-1} w_i \text{ for } x \in X \quad (4)$$

여기서 N 은 분할된 총 블록 수이다. 원영상과 끌개 사이의 거리를 최소로 하는 변환을 찾는 것은 힘들기 때문에 실제로는 콜라쥬 정리가 사용된다. 삼각 부등식과 축소변환의 정의에 의해 그 정리는 다음과 같이 표현된다.

$$d(x, x_f) \leq \frac{1}{1-s} d(x, w(x)) \quad (5)$$

식(5)는 원영상 x 가 주어졌을 때 x 에 충분히 가까운 콜라쥬 영상을 생성하는 변환 w 는 그 변환의 끌개 또한 x 에 가까움을 의미한다. 그래서 이 정리는 원영상과 끌개 사이의 거리를 최소로 하는 변환을 구하는 대신 원영상과 콜라쥬 영상 사이의 거리

를 최소로 하는 변환을 찾음으로써 부호화 부담을 덜어준다.

프랙탈 부호화에 있어서 가장 널리 사용되는 변환은 다음식에 나타나는 어파인 함수이다.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = w_i \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \\ I(x, y) \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta g \end{bmatrix}, \quad (6)$$

여기서 I 와 J 는 정의역 블록과 치역 블록의 밝기 값이고 (a, b, c, d) 는 기하학적 변환(geometric transformation)과 셔플 변환(shuffle transformation)을 나타내는 계수들이고, $(x, y, \Delta g)$ 는 변위 벡터로서 (x, y) 는 공간적 변위향으로서 정의역 블록의 위치를 나타내고, Δg 는 밝기 값의 변위향으로서 정의역 블록과 치역 블록의 밝기 값의 차이를 나타낸다.

프랙탈 부호화 방식들은 일반적으로 원영상을 접치지 않는 블록들로 나누고, 그 블록에 대한 변환 w_i 를 전체 영상 또는 영상의 일부분에 대해 정의역 블록을 찾는 과정으로 이루어진다. 이 때 나누어진 그 블록을 치역 블록이라고 부르고 치역 블록에 대한 변환은 다음 식을 만족하도록 찾는다.

$$\min_{a, b, c, d, s, \Delta x, \Delta y, \Delta g} d(R, w_i(D)) \quad (7)$$

여기서 R 과 D 는 각각 치역 블록과 정의역 블록이다. 식(7)의 각 계수들은 미리 정해진 간격으로 양자화되고 양자화된 모든 값에 대해 변환된 값, $w_i(D)$ 와 치역 블록 R 을 비교해서 거리가 가장 작은 변환을 선택한다. 이처럼 w_i 를 찾는 것은 미리 정해진 변환에 의한 블록 $w_i(D)$ 와 R 과의 정합 오차가 작도록 하는 변환을 선택하는 것이다. 여기서 $w_i(D)$ 는 콜라쥬 블록이라고 불리고 앞으로 언급되는 정합 오차는 $w_i(D)$ 와 R 과의 오차를 의미한다.

II. 물체 기반 프랙탈 부호화

일반 영상에 대한 프랙탈 부호화 방식은 영상을 접치지 않는 블록들로 나눈 뒤, 그 블록을 치역 블록으로 하여 유사한 모양의 정의역 블록을 찾고 부호화한다. 물체 단위의 프랙탈 부호화 방식도 마찬가지로 구현될 수 있다. 물체 기반 프랙탈 부호화에서 치역 블록은 배경 영역과 물체 영역의 포함에 따라 두개의 부류로 나눌 수 있다. 즉, 그림 1에서 보듯이 배경과 물체가 공존하는 경계 블록(boundary

block)과 물체만 존재하는 내부 블록(internal block)로 나누어진다. 그리고 배경 영역만 존재하는 블록들도 생각해 볼 수 있지만 이 블록들은 부호화가 필요 없는 블록이다.

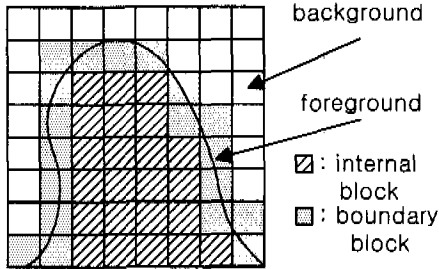


그림 1. 물체의 실례

전체 영상을 $b \times b$ 크기의 겹치지 않는 블록들로 나누었을 때 그 블록 $B_{i,j}$ 은 다음과 같이 표현된다.

$$B_{i,j}(m, n) = I(i \times b + m, j \times b + n),$$

$$0 \leq m < b, 0 \leq n < b, 0 \leq i < N_x, 0 \leq j < N_y \quad (8)$$

여기서 $I(x,y)$ 는 (x,y) 에서의 밝기 값이고 N_x, N_y 는 x 방향의 블록수, y 방향의 블록수이다. 블록 $B_{i,j}$ 은 다음과 같은 마스크를 정의함으로써 내부 블록과 경계 블록, 그리고 배경 블록 중의 하나로 분류될 수 있다.

$$M_{i,j}(m, n) = \begin{cases} 1, & (i \times b + m, j \times b + n) \in \text{Foreground} \\ 0, & (i \times b + m, j \times b + n) \in \text{Background} \end{cases},$$

$$0 \leq m < b, 0 \leq n < b \quad (9)$$

$M_{i,j}(m, n)$ 의 모든 값이 1인 경우 ($\sum_{m,n} M_{i,j}(m, n) = b^2$)는 내부 블록에 해당되고, 모두 0인 경우 ($\sum_{m,n} M_{i,j}(m, n) = 0$)는 배경 블록, 그리고 0과 1이 함께 존재하는 경우 ($0 \leq \sum_{m,n} M_{i,j}(m, n) < b^2$)는 경계 블록이다. 이 때 내부 블록의 집합을 S_I , 경계 블록의 집합을 S_B 라고 하자. 이 마스크를 이용하면서 지역 블록 $R_{i,j}$ 는 다음과 같이 정의된다.

$$R_{i,j}(m, n) = B_{i,j}(m, n) \times M_{i,j}(m, n). \quad (10)$$

또한 $d \times d$ 크기의 정의역 블록은 다음과 같이 정

의될 수 있다.

$$D_k(m, n) = I(k \times \Delta_s + m, l \times \Delta_s + n),$$

$$0 \leq m < d, 0 \leq n < d, \quad (11)$$

여기서 Δ_s 는 정의역 블록을 찾는 간격을 의미한다. 일반적으로 Δ_s 이고, 이 경우 $0 \leq k < N_x - 1, 0 \leq l < N_y - 1$ 이다. 배경 블록은 부호화할 필요가 없으므로 부호화할 필요가 없고 내부 블록과 경계 블록은 부호화되어야 한다. 내부 블록들에 대한 부호화는 표준 부호화 방식을 따르면서 쉽게 부호화될 수 있지만 경계 블록은 배경이 함께 존재하는 블록이므로 유효 화소의 수가 내부 블록에 비해 적으므로 내부 블록과는 다른 처리가 요구된다. 그래서 이 논문에서는 경계 블록의 부호화 방식에 대해 논의를 집중하고 기술한다. 일반적인 영상에 대한 프랙탈 부호화 방식과 마찬가지로 공간 영역에서의 정합에 의한 방법과 주파수 영역에서의 정합에 의한 방법이 있을 수 있다. 그리고 경계 블록의 모양이 사각형이 아니므로 이를 사각형으로 만들어 정합하는 방법을 생각할 수 있다. 또한 경계 블록의 정의역은 내부 블록들로 구성되는 특성을 고려함으로써 경계 블록의 복원 화질을 개선하는 방법을 제안한다.

1. 공간영역에서의 정합에 의한 방법

보통의 영상에 적용되는 방법처럼 임의의 모양을 가지는 지역 블록과 같은 모양의 블록을 생각할 수 있다. 그 블록은 정의역 블록이라고 불린다. 지역 블록의 마스크가 정의된 것처럼 정의역 블록의 마스크가 다음과 같이 정의될 수 있다.

$$F_k(m, n) = M_{i,j}(m/2, n/2), 0 \leq m < d, 0 \leq n < d, \quad (12)$$

여기서 마스크 $F_k(m, n)$ 는 $R_{i,j}(m, n)$ 의 마스크 $M_{i,j}(m, n)$ 가 두 배로 확대된 것이다. 그때 마스크된 정의역 블록은 다음과 같이 표현될 수 있다.

$$D_k^F(m, n) = D_k(m, n) \times F_k(m, n),$$

$$0 \leq m < d, 0 \leq n < d. \quad (13)$$

하나의 지역 블록에 대한 변환을 얻기 위해서는 정의역 풀 내에 있는 정의역 블록들에 대한 서플 변환과 기하학적 변환을 수행해야 한다. 그러나 마스크된 정의역 블록 $D_k^F(m, n)$ 의 기하학적 변환과 서플 변환은 불편하다. 그것은 임의의 모양을 가진

블록의 회전과 대칭이동은 원래의 모양을 보존하지 못하기 때문이다. 예를 들어 폭이 높ی보다 큰 직사각형 블록의 90° 회전은 높이가 폭보다 큰 직사각형 블록으로 변한다. 그러므로 일반적으로 서플 변환된 정의역 블록과 치역 블록의 모양이 서로 다르므로 정합이 불가능하다. 그래서 우리는 $F_{k,i}(m, n)$ 을 $D_{k,i}(m, n)$ 에 적용하는 대신 $D_{i,j}(m, n)$ 을 변환된 블록에 적용한다. 그때 치역 블록과 정합되는 변환된 정의역 블록은 다음과 같다.

$$E_{k,i}(m, n) = S_p(G(D_{k,i}(m, n))) \times M_{i,j}(m, n),$$

$$0 \leq m < b, 0 \leq n < d, 0 \leq m < d, 0 \leq n < a \quad (14)$$

여기서 $G(\cdot)$ 는 기하학적 변환이고, $S_p(\cdot)$ 는 p 번째의 서플 변환을 의미한다. 일반적으로 서플 변환은 8가지이므로 $0 \leq p < 8$ 이다^[2]. 밝기 값에 대한 축소 변환을 $E_{k,i}$ 에 적용하면 최종적으로 얻어지는 변환된 정의역 블록은 다음과 같이 주어진다.

$$u(D_{k,i}) = \lambda E_{k,i}(m, n) + \Delta g \quad (15)$$

여기서 λ 는 밝기 값에 대한 축소 계수이고 Δg 는 밝기 값에 대한 변위항이다. 그래서 변환 계수들은 다음과 같은 식에 의해 구해진다.

$$\min_{k,i,p,\lambda,g,\Delta g} \sum_{(m,n) \in M} (R_{i,j}(m, n) - \lambda E_{k,i}(m, n) + \Delta g)^2 \quad (16)$$

그림2에서 보는 것처럼 배경 영역에 속한 화소들은 정합 과정에서 제외되고 물체 내부의 화소들만 정합 과정에서 고려된다. 이 방식은 모양이 사각이 아니라는 점을 제외하면 보통의 프랙탈 부호화 방식과 거의 같다.

이 방식에 의해 발생한 평균 정합 오차는 정합해야 할 화소의 수가 적으므로 내부 블록들의 부호화 과정에서 발생하는 정합 오차와 비교하여 작아질 것으로 예상할 수 있지만 물체의 경계 부분의 화소의 밝기 값의 변화가 심하기 때문에 오히려 정합 오차가 커질 수 있다. 이 절에서 설명한 이 방식은 앞으로 방식1이라고 불린다.

2. 주파수영역에서의 정합에 의한 방법

앞 절에서 보았듯이 임의의 모양을 가진 블록에 대한 정합이 주파수 영역에서도 수행될 수 있다. 신호의 파워는 공간 영역과 주파수 영역에서 보존되기 때문에 주파수 영역에서의 정합에 의한 방법은

공간 영역에서의 정합에 의한 방법으로 얻어진 결과와 같다. 즉, 주파수 영역에서 방식1을 구현하는 것은 같은 결과의 변환을 얻으므로 구현의 복잡도만 증가될 뿐 장점이 없다. 그러나 이 절에서 제안된 방식은 임의의 모양을 가진 치역 블록과 사각 모양의 정의역 블록과의 주파수 영역에서의 정합을 수행하는 방식으로서 방식1과는 다른 결과를 준다. 이 방식은 치역 블록의 복잡도가 커서 어파인 함수로 잘 근사화될 수 없어서 정합 오차가 큰 경우에 장점을 가진다. 다음 장에서 이 방식에 대한 분석을 통해 우리는 그 장점을 보일 것이다.

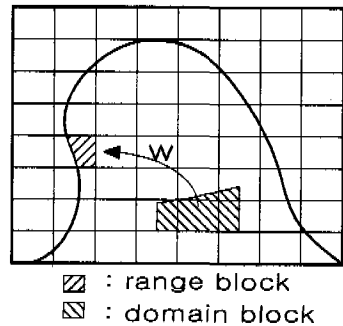


그림 2. 공간영역에서의 정합에 의한 방법

우선 주파수 영역에서의 정합을 위해서는 치역 블록과 정의역 블록을 주파수 영역으로 변환하는 과정이 필요하다. 임의의 모양을 가지는 치역 블록을 주파수 영역으로 변환하는 방식으로는 크게 두 개의 부류로 나눌 수 있다. 그 하나는 임의의 모양을 가진 블록을 사각 블록으로 변형한 후 사각 블록이 주파수 영역으로 변환되고 그 때 그 주파수 성분을 임의의 모양을 가진 블록의 주파수 성분으로 간주하는 방법이다. mirror extension 방식, mean stuffing 방식 그리고 EI방식이 이 부류에 속한다. mirror extension 방식과 mean stuffing 방식은 배경 영역의 화소, 즉, 블록 내의 유효하지 않은 화소들을 물체 영역의 화소, 즉, 블록 내의 유효한 화소들을 이용하여 채운다^[18]. mirror extension 방식과 mean stuffing 방식은 원래 신호에 존재하지 않는 고주파 성분이 발생시키는 반면 EI방식은 그렇지 않다. EI방식에 대한 자세한 설명은 참고문헌[14]와 [15]에 있고 EI방식을 이용한 경계 블록에 대한 프랙탈 부호화 방식은 다음 절에서 설명된다. 그리고 두번째 부류는 블록의 모양을 유지하면서 주파수 성분들은 얻는 부류이다. SADCT(shape adaptive

DCT), GOT(generalized orthogonal transformation)가 이 부류에 속한다^{[16][17]}.

우리는 임의의 모양을 가진 블록을 주파수 영역으로 변환하는데 널리 사용되는 SADCT를 변환도구로 사용하였다. SADCT를 사용하면서 변환된 치역 블록은 다음과 같이 표현된다.

$$M'_{i,j}(u, v) = SADCT[R_{i,j}(m, n)],$$

$$(u, v) \in M'_{i,j}, (m, n) \in M_{i,j} \quad (17)$$

여기서 $M'_{i,j}$ 는 주파수 영역에서의 마스크이고 $M_{i,j}$ 은 공간 영역에서의 마스크이다. 그림3는 치역 블록을 SADCT하는 과정으로써 공간 영역과 주파수 영역의 마스크 $M_{i,j}$ 와 $M'_{i,j}$ 를 보여준다.

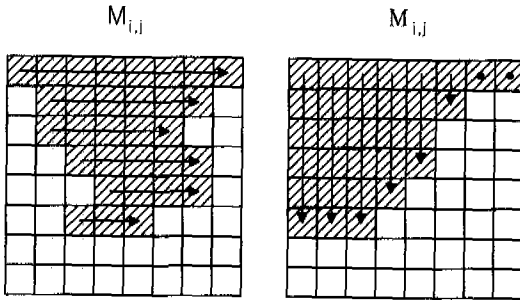


그림 3. SADCT와 공간 영역과 주파수 영역에서의 마스크들

한편 정의역 블록의 모양은 두 부류로 분류해 볼 수 있다. 그 하나는 치역 블록의 모양과 같은 부류이고 나머지 하나는 치역 블록의 모양에 상관없이 사각 모양인 부류이다. 치역 블록과 같은 모양의 정의역 블록의 사용은 방식1과 같은 결과를 가지므로 이 절에서는 정의역 블록이 사각 블록인 경우를 다룬다.

사각 모양의 정의역 블록의 주파수 성분은 2-D DCT를 수행함으로써 쉽게 얻을 수 있고 이를 치역 블록의 SADCT 과정에서 얻어진 마스크 $M'_{i,j}$ 를 사용하여 마스크함으로써 치역 블록과의 정합이 가능하다. 치역 블록과 정합을 위한 주파수 영역에서의 정의역 블록은 다음과 같이 주어진다.

$$D_{k,l}(u, v) = DCT[S_p(D_{k,l}(m, n))]M'_{i,j}(u, v),$$

$$0 \leq u < b, 0 \leq v < b, 0 \leq m < d, 0 \leq n < d, \quad (18)$$

여기서 DCT된 정의역 블록에 대한 서플 변환은 DCT 계수의 부호의 변경과 대칭이동에 의해 쉽게 구현될 수 있다. 식(17)과 (18)로부터 변환 계수는 다음 식에 의해 얻어진다.

$$\min_{k,l,p,\lambda_g, \Delta g} \sum_{(u,v) \in M'_{i,j}} (M'_{i,j}(u, v) - \lambda_g D'_{k,l}(u, v) - \Delta g \delta(u, v))^2 \quad (19)$$

여기서 $\delta(u, v)$ 는 kronecker delta 함수이다. 이 방식의 수행 과정이 그림4에 보여진다.

이 방식을 주파수 영역 대신 공간 영역에서 구현하는 방법을 생각해 보자. 주파수 영역에서 정의역 블록을 마스크하는 것은 공간 영역에서 필터링(filtering)과 간축(decimation)함으로써 치역 블록의 모양과 같도록 하는 것과 같다. 그래서 이 방식은 사각형의 정의역 블록을 치역 블록과 같은 모양을 갖도록 한 후, 정합을 수행하여 변환 계수를 얻는 방식이다. 이러한 사실로부터 방식1과 비교하여 이 방식은 공간적 축소성이 좀 더 크음을 알 수 있다. 이 절에서 소개된 방식은 앞으로 방식2라고 불린다.

3. 주파수영역에서의 정합에 의한 방법

방식2가 정의역 블록을 치역 블록의 모양에 맞추어 정합하는 방식이라면 이 절에서 소개되는 방식은 치역 블록을 정의역 블록의 모양에 맞추어 정합하는 방식이다. 즉, 임의의 모양을 가지는 치역 블록을 EI를 적용하여 사각 블록으로 만든 후, 그 블록과 유사한 정의역 블록을 찾는 방법이다. EI는 임의의 모양을 가진 블록을 부호화하기 위해 제안된 방법으로서 mirror extension 방식과 mean stuffing 방식과는 달리 원래 신호가 가지고 있던 주파수 성분 외의 고주파 성분들을 억제하면서 사각블록으로 만드는 방법이다. EI를 수행함으로써 임의의 모양을 가지는 치역 블록은 다음과 같은 사각 블록로 변형된다.

$$R'_{i,j}(m, n) = EI[R_{i,j}(m, n)], 0 \leq m < b, 0 \leq n < b \quad (20)$$

그리고 정의역 블록은 축소 변환에 의해 다음과 같이 얻어진다.

$$u(D_{k,l}) = \lambda_g S_p(G(D_{k,l}(m, n))) + \Delta g,$$

$$0 \leq m < d, 0 \leq n < a \quad (21)$$

식(20)과 (21)로부터 다음 식을 만족하는 변환 계수를 찾는다.

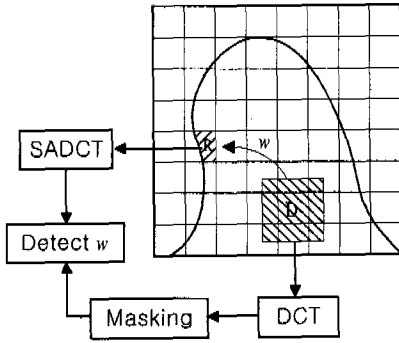


그림 4. 주파수영역에서의 정합에 의한 방법

$$\min_{k, l, p, \lambda, \Delta E} \sum_{m=0}^{k-1} \sum_{n=0}^{l-1}$$

$$(R_{i,j}^{EI}(m, n) - w(D_{k,l})(m, n))^2 \quad (22)$$

EI를 이용한 방식은 지역 블록의 모양을 사각 블록으로 변형했을 때, 그 후의 모든 과정은 보통의 프랙탈 부호화 방식과 같은 과정으로 그 변환 계수들을 찾는다. 여기서 식(22)에 의해 얻어진 변환 계수들은 $R_{i,j}$ 와 $D_{k,l}$ 에 대한 것이 아니라 $R_{i,j}^{EI}$ 와 $D_{k,l}$ 에 대한 것임을 유의해야 한다. 그래서 복호화 과정에서 방식1과 방식2와는 달리 지역 블록을 다시 원래 윤곽선 모양에 맞추기 위해서 EI의 역과정이 필요하다. 그래서 최종적으로 복호화된 블록은 다음과 같다.

$$\widehat{R}_{i,j}(m, n) = IEI[R_{i,j}^{EI}(m, n)], 0 \leq m < b, 0 \leq n < b \quad (23)$$

여기서 IEI는 EI의 역과정이고, $R_{i,j}^{EI}(m, n)$ 는 식(22)를 만족하는 변환에 의해 복호화된 사각 블록이다.

우리는 이 방식을 방식3이라고 부르고 그림5에서 보여지는 것처럼 다음과 같은 과정으로 구현된다.

- (1) EI방식을 사용하면서 경계 블록 $R_{i,j}$ 에 대한 $R_{i,j}^{EI}$ 를 얻어라.
- (2) $R_{i,j}^{EI}$ 의 정의역 블록을 찾고 정의역 블록을 $R_{i,j}^{EI}$ 로 변환하는 변환 계수를 결정하라
- (3) 모든 경계 블록들에 대해 위의 과정을 수행하고 변환 계수들을 부호화하라.

방식1과 방식2와는 달리 방식3의 경계 블록들은 복호화시 다음과 같은 추가적인 처리가 필요하다.

- (1) 수신된 변환 계수들로부터 반복적인 변환을 수행하면서 모든 경계 블록에 대해서 $R_{i,j}^{EI}$ 를 얻어라.
- (2) 식(23)처럼 IEI를 수행함으로써 최종적으로 $\widehat{R}_{i,j}$ 를 얻어라.

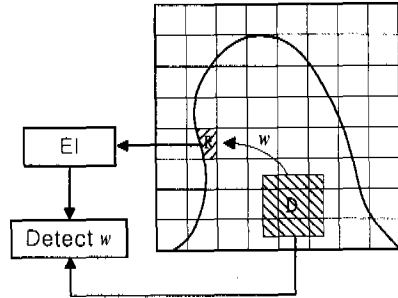


그림 5. EI방식을 이용한 지역 블록의 확정에 의한 방법

4. 내부블록의 선부호화와 경계블록의 후부호화에 의한 방식 (FILB)

물체 기반 프랙탈 부호화에서 정의역 풀(pools)로 사용되는 영역은 물체 내부이다. 물체에 대한 우리의 부호화 방식에서 물체는 블록 단위로 분할되고 그 블록들에 대한 변환을 찾으므로 물체의 경계 부분이 정의역 블록으로 사용되기에는 경계 부분의 모양이 불규칙적이고 고주파 성분이 많이 존재한다. 즉, 경계 영역을 정의역 블록으로 사용할 경우 지역 블록과의 정합을 위한 화소의 수가 부족하기 때문에 경계 부분은 정의역으로 선택되지 않는다. 또한 물체 내부에 비해 경계 영역이 큰 비중을 차지하지 못하므로 물체 경계에 해당되는 블록을 정의역 풀에서 제외하더라도 부호화기의 성능에 영향을 거의 미치지 못한다. 그리고 경계 부분은 배경과 물체의 밝기 값이 공존하기 때문에 대체로 고주파 성분이 많이 존재함으로써 정의역 풀로 사용되기에는 부적합하다. 그래서 경계 영역을 인위적으로 정의역 풀에서 제외시키지 않더라도 많은 고주파 성분과 화소의 부족으로 저절로 제외된다. 이러한 사실로부터 우리의 물체 기반 프랙탈 부호화기에서는 경계 부분의 블록들은 정의역 풀에서 제외된다. 즉, 정의역 풀은 지역 블록이 경계 블록인지 내부 블록인지에 상관없이 내부 블록들로만 구성된다. 그래서 경계 블록에 속하는 지역 블록들은 경계 블록의 정의역 블록이 경계 블록들로 구성되는 것이 아니라 내부

블록들로 구성되므로 복호화된 내부 블록들을 단 한번 변환함으로써 복원이 가능하다. 이러한 특성은 미리 복호화된 부분을 정의역으로 사용함으로써 경계 블록에 대한 프랙탈 부호화기 성능을 향상시키는 데 도움을 준다. 즉, 내부 블록들에 대한 프랙탈 변환들을 우선적으로 찾고 그 변환들을 이용하여 미리 내부 블록들을 복호화한 결과를 원영상 대신 경계 블록들의 정의역으로 사용하면 경계 블록에 대한 복원 화질이 개선될 수 있다. 이처럼 블록들의 부호화 순위를 인위적으로 조작하는 것은 추가적인 정보의 전송을 의미하지 않는다. 왜냐하면 물체의 윤곽선 정보로부터 그 순위를 알 수 있을 뿐만 아니라 복호화시 그 순위에 상관없이 복호화하더라도 고정점 정리(fixed point theorem)에 의해 같은 결과를 얻기 때문이다. 내부 블록들을 우선적으로 부호화하고 미리 복호화한 후, 경계 블록에 대한 변환을 다음 식에 의해 얻는다.

$$\min_{k, l, p, \lambda, \Delta G} \left(\sum_{(m,n) \in M} R_{i,j}(m,n) - w(D_{k,l})(m,n) \right)^2, \quad \forall R_{i,j} \in S_B \quad (24)$$

여기서 $\widehat{D}_{k,l}$ 는 내부 블록들이 미리 복원된 후의 값으로 구성된 정의역 블록이고 w 는 $\widehat{D}_{k,l}$ 와 $R_{i,j}$ 의 거리를 최소화 하는 변환이다. 식(24)과 비교하여 다음 식처럼 일반적인 부호화 방식에서는 정의역으로써 원영상이 사용된다.

$$\min_{k, l, p, \lambda, \Delta G} \left(\sum_{(m,n) \in M} R_{i,j}(m,n) - w(D_{k,l})(m,n) \right)^2, \quad \forall R_{i,j} \in S_B \quad (25)$$

이제 정의역으로 식(25)처럼 원영상을 사용한 경

우와 식(24)처럼 복원된 영상을 사용하는 경우를 비교해 보자. 부호화하고자 하는 영상을 x^b 라고 할 때, 경계 블록으로 구성된 영상 x^b 와 내부 블록들로만 구성된 영상 x^i 로 나누어 볼 수 있다. 마찬가지로 변환에 의해 복원된 영상 x_r 는 경계 블록들의 복원된 영상 x_r^b 과 내부 블록들의 복원된 영상 x_r^i 로 나누어 볼 수 있다. 식(24)을 이용하여 얻어진 x^b 에 대한 변환을 w_a 라고 하고 식(25)을 이용하여 얻어진 x^b 에 대한 변환을 w_b 라고 하자. 이 때 경계 영역에 대한 복원 오차는 각각 $d(x^b, w_a(x_r^i))$, $d(x^b, w_b(x_r^i))$ 이다. 여기서 x_r^i 는 내부 블록들의 정의역은 내부 블록들로 구성되어 있기 때문에 경계 영역의 복원 결과와는 독립적으로 내부 블록들에 의해 결정된다. 각각의 복원 오차는 식(24)에 의해 다음과 같은 관계를 갖는다.

$$d(x^b, w_a(x_r^i)) \leq d(x^b, w_b(x_r^i)) \quad (26)$$

식(24)에서 w_a 는 w_b 와 x_r^i 사이의 거리를 최소로 하는 변환이기 때문에 식(26)은 명확하게 성립한다. 그래서 w_a 는 w_b 보다 좀 더 작은 최대 한계를 갖는다. 결과적으로 경계 영역에 대한 프랙탈 변환을 찾을 때는 식(25)을 이용하는 대신 식(24)을 이용하는 것이 복원 화질을 개선한다. 식(24)에 기초한 제안된 방식은 FILB방식이라고 이름짓고 다음과 같은 과정을 따라 물체 기반 프랙탈 부호화를 수행한다.

- (1) 물체를 접치지 않는 블록으로 나눈 후, 경계 블록과 내부 블록으로 분류한다.

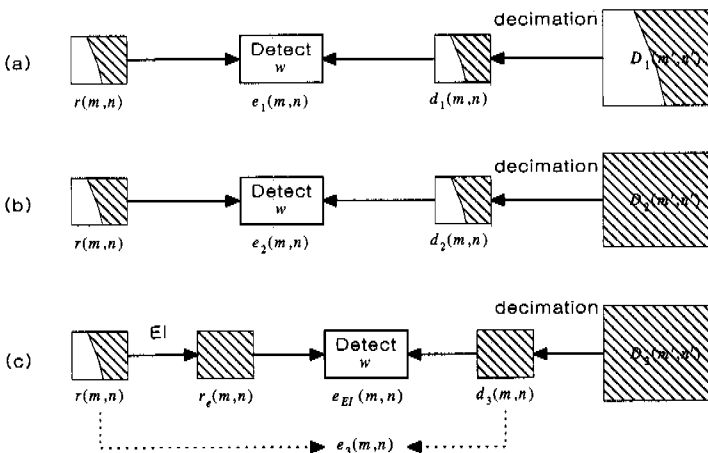


그림 6. 제안된 방법들의 요약

- (a) 공간 영역에서의 정합에 의한 방법
- (b) 주파수 영역에서의 정합에 의한 방법
- (c) EI 방식을 이용한 방법

- (2) 내부 블록들에 대한 변환을 표준 풀라쥬 부호화 기법을 이용하여 모두 찾는다.
- (3) 내부 블록들에 대한 변환으로부터 반복 변환에 의해 미리 복호화하고 내부 블록들을 복호화된 결과로 대치한다.
- (4) 경계 블록들에 대한 변환을 찾고 부호화 한다. 이 때 경계 블록들의 정의역 블록은 과정3에 의해 원영상이 아닌 복호화된 결과로 구성되어 있다.

이 절에서 우리는 방식1, 방식2, 또는 방식3와 같이 경계 블록에 속하는 치역 블록에 대한 부호화 방식을 자세히 언급하지 않았다. 단지 FILB방식은 물체 기반 프랙탈 부호화의 성능 개선을 위한 블록의 부호화 순위에 대한 것이다. 앞 절에서 언급된 방식1, 방식2, 그리고 방식3들 모두가 FILB방식과 동시에 적용될 수 있고 이것은 그 방식들이 단독으로 사용되는 것보다 우수한 성능을 보인다.

IV. 제안된 방식들의 분석

이 장에서는 3장에서 언급된 3가지 방식의 성능을 분석한다. 여기서 방식1은 방식2와 방식3의 성능 평가를 위한 기준으로 사용된다. 각 방식들에 대한 요약이 그림 6에 보여진다. 우선 각 방식에 의해 얻어지는 경계 블록들의 복원 화질에 대해 살펴보자. 수식 전개의 편의를 위해 프랙탈 변환을 다음과 같이 표현하자.

$$w(x) = Lx + b \tag{27}$$

여기서 L 은 서플 변환과 기하학적 변환을 표현하는 행렬이고 b 는 변위 벡터이다. 식(27)을 경계 블록에 대해 쓰면 다음과 같다.

$$\hat{x}^b = w_B(x^i) = L_B x^i + b_B \tag{28}$$

여기서 x^i 는 내부 블록으로 구성된 영상 벡터이고 L_B 와 b_B 는 경계 블록들에 대한 서플 변환과 기하학적 변환이다. 식(28)은 경계 블록들의 정의역 풀은 내부 블록들로 구성된다는 사실로부터 주어진다. 변환 w_B 는 풀라쥬 정리에 의해 x 와 \hat{x}^b 의 거리를 최소로 하도록 정해진다. 그러나 복호화기에서 최종적으로 얻어지는 경계 블록들의 복원 결과는 다음과 같다.

$$\hat{x}_f^b = w_B(x_f^i) = L_B x_f^i + b_B \tag{29}$$

여기서 x_f^i 는 복원된 내부 블록들로 구성된 벡터이다. 식(29)는 경계 블록의 정의역 풀이 내부 블록들로 구성되므로 내부 블록들이 복호화되면 그 복호화된 결과를 한번 변환한 것이 경계 블록의 끝개가 됨을 보여준다. x_f^i 를 원영상 x^i 과 오차신호 e 의 합으로 표현하면, 즉, $e = x_f^i - x^i$, 식(29)는 다음과 같이 다시 쓸 수 있다.

$$\begin{aligned} \hat{x}_f^b &= L_B(x^i + e) + b_B = L_B x^i + b_B + L_B e \\ &= \hat{x}^b + L_B e \end{aligned} \tag{30}$$

식(30)으로부터 복원 오차는 다음과 같이 그 크기가 한정지어진다.

$$\|x^b - \hat{x}_f^b\| \leq \|x^b - \hat{x}^b\| + \|L_B e\| \tag{31}$$

여기서 $\|\cdot\|$ 는 벡터 또는 행렬의 norm로서 식(1)의 축소 계수 λ 에 해당된다. 식(31)으로부터 경계 영역의 복원 화질은 경계 블록과 그것의 풀라쥬 블록 사이의 거리 $\|x^b - \hat{x}^b\|$ 와 내부 블록의 부호화에 의한 복원 오차 $\|e\|$ 와 축소 계수 $\|L_B\|$ 의 곱으로 주어짐을 알 수 있다. $\|L_B\|$ 이 작을 수록 식(31)의 우변의 값이 작아지므로 성능 개선을 기대할 수 있지만 작은 값의 $\|L_B\|$ 은 오히려 $\|x^b - \hat{x}^b\|$ 를 증가시키는 결과를 초래할 수 있다. 즉, 작은 값의 $\|L_B\|$ 은 공간적 축소율(정의역 블록의 크기 대 치역 블록의 크기)을 크게 하거나 기하학적 변환(공간적 축소자)에 있어서 필터의 차단 주파수를 낮춤으로써 얻어진다. 그래서 공간적 축소자의 차단 주파수를 낮추거나 좀 더 큰 크기의 정의역 블록의 사용은 정합 오차 $\|x^b - \hat{x}^b\|$ 를 크게 만들 수 있으므로 유의해야 한다. 그러나 물체의 경계 부분처럼 $\|e\|$ 가 큰 경우에는 작은 값의 $\|L_B\|$ 은 복원화질 개선에 도움을 줄 수 있다.

이제 식(31)에서 주어진 경계 블록들에 대한 복원 화질을 이용하여 방식1과 방식2를 비교해 보자. 3장에서도 언급했듯이 방식2의 정의역 블록은 방식1의 정의역 블록에 비해 크기가 크므로 $\|L_B\|$ 가 작은 값을 가진다. 그리고 오차신호 e 는 내부 블록들의 복원 오차이므로 방식1과 방식2의 e 는 같기 때문에 식(31)의 우변의 둘째 항이 방식2가 방식1에 비해

작아진다. 즉, 방식2가 방식1에 비해 우수한 복원 화질을 가질 수 있음을 의미한다. 그러나 작은 값의 $\|L_{E1}\|$ 는 정합오차 $\|x^b - \hat{x}^b\|$ 의 증가를 의미하므로 방식2가 방식1에 비해 우수하다고 단정할 수 없다. 그러나 오차신호 e 가 주도적인 경우, 작은 값의 $\|x^b - \hat{x}^b\|$ 은 총 오차를 줄이는 효과가 있을 수 있다. 그러므로 방식2는 방식1에 비해 우수한 성능을 가질 수 있다. 이러한 사실을 우리는 다음 장에서 실험적으로 방식2가 오차신호가 큰 경우에 방식1에 비해 좀 더 우수한 성능을 가짐을 보일 것이다.

이제 FILB를 방식2와 함께 적용했을 경우에 대해 생각해 보자. FILB는 식(28)에서 보듯이 정의역 M 으로서 x^i 대신 x^j 를 사용하는 방식이다. 식(28)에 x^i 대신 x^j 를 대입함으로써 식(31)은 $\|x^b - \hat{x}^b\| = \|x^b - \hat{x}^b\|$ 가 된다. 이것은 경계 블록이 복원된 내부 블록을 한번 변환함으로써 복원될 수 있음을 말한다. 방식2는 $\|L_{E1}\|$ 를 감소시키는 효과에 의해 성능 개선의 여지가 있었지만 이 방식을 함께 사용함으로써 방식2의 사용은 무의미해지고 오히려 복원 오차를 증가시킬 수 있다. 그래서 FILB와 방식2를 함께 적용하는 것은 FILB만을 적용한 경우와 비슷한 결과이거나 오히려 나쁜 결과를 얻을 수 있다. 여기서 FILB만을 적용한 것이라 FILB와 방식1을 적용한 경우이다.

지금까지 우리는 방식1과 방식2를 비교하였다. 이제 방식1과 방식3을 비교해 보자. 우선 그림6에 나타난 것처럼 각 방식의 치역 블록과 그것의 플라워 블록과의 정합 오차 신호를 다음과 같이 정의하자.

$$e_i(m, n) \equiv r(m, n) - \hat{r}_i(m, n), i = 1, 2,$$

$$e_e(m, n) \equiv r(m, n) - IEI[\hat{r}_3(m', n')],$$

where $\hat{r}_i(m, n) = u(d_i)$,

$$(m, n) \in M, 0 \leq m' < b, 0 \leq n' < b \quad (32)$$

여기서 M 은 물체 영역에 대한 마스크이다. 이 식에서 보듯이 방식3의 경우 방식1과 방식2와는 달리 IEI의 과정이 추가적으로 필요함에 유의해야 한다. 그래서 중간 단계의 오차 $e_{EI}(m', n')$ 를 다음과 같이 정의한다.

$$e_{EI}(m', n') \equiv r_{EI}(m', n') - \hat{r}_3(m', n') \quad (33)$$

여기서 $e_{EI}(m', n')$ 는 $r_{EI}(m', n')$ 와 $\hat{r}_3(m', n')$ 의 차이 신호인 반면 $e_3(m, n)$ 는 $r(m, n)$ 와 $IEI[\hat{r}_3(m', n')]$ 의 차이 신호이다. 즉, $e_{EI}(m', n')$ 는 $b \times b$ 블록 전체 영역에 대해 유효한 반면 $e_3(m, n)$ 는 M 내에서만 유효하다. 식(33)에서의 정의와 함께 $IEI[\hat{r}_3(m', n')]$ 는 다음과 같이 표현될 수 있다.

$$\begin{aligned} IEI[\hat{r}_3(m', n')] &= IEI[r_{EI}(m', n') - e_{EI}(m', n')] \\ &= IEI[r_{EI}(m', n')] - IEI[e_{EI}(m', n')] \\ &= r(m, n) - IEI[e_{EI}(m', n')] \end{aligned} \quad (34)$$

식(34)를 식(32)에 대입하면 우리는 $e_3(m, n) = IEI[e_{EI}(m', n')]$ 의 관계식을 얻는다. 그때 $e_3(m, n)$ 의 평균자승값(mean square value)은 다음과 같다.

$$\begin{aligned} \sigma_3^2 &= \sum_{(m, n) \in M} |e_3(m, n)|^2 \\ &= \sum_{(m, n) \in M} |IEI[e_{EI}(m', n')](m, n)|^2 \\ &= \sum_{(u, v) \in M'} |E_{EI}(u, v)|^2, 0 \leq m' < b, 0 \leq n' < b \end{aligned} \quad (35)$$

여기서 M' 은 $r(m, n)$ 의 주파수 영역에서의 마스크이다. $e_{EI}(m, n)$ 의 평균자승값 σ_{EI}^2 라고 할 때 σ_3^2 와 σ_{EI}^2 의 비는 다음과 같다.

$$\begin{aligned} \frac{1}{p^2} &\equiv \frac{\sigma_3^2}{\sigma_{EI}^2}, \\ &= \frac{\sum_{(u, v) \in M'} |E_{EI}(u, v)|^2}{\sum_{(u, v) \in M'} |E_{EI}(u, v)|^2 + \sum_{(u, v) \in (M')^c} |E_{EI}(u, v)|^2} \end{aligned} \quad (36)$$

여기서 $(M')^c$ 은 M' 의 여집합이다. p 가 커질수록 σ_3^2 는 σ_{EI}^2 에 비해 상대적으로 작아짐을 알 수 있다. p 가 크다는 것은 마스크 M' 의 크기가 작을 때이므로, 즉 $\sum M'$ 가 작은 값을 가질 때이므로 $e_{EI}(m, n)$ 가 고주파 성분이 적음을 의미한다. 예를 들어 $e_{EI}(m, n)$ 가 백색잡음(white noise)이고 $a^2 = \sum M'$ 이면 $p = b/a$ 이므로 $\sigma_3^2 = a^2 \sigma_{EI}^2 / b^2$ 이다. 여기서 b 는 식(8)에 나타난 블록의 크기이다. 그리고 $a/b \leq 1$ 이므로 σ_3^2 는 항상 σ_{EI}^2 보다 작거나 같다. 마스크 M' 의 크기가 작아질 수록 a/b 의 값이 작아지므로 σ_3^2 는 작아진다. 그래서 IEI는 $r_{EI}(m, n)$ 와

$\hat{r}_3(m, n)$ 사이의 정합 오차를 줄이는 효과가 있음을 알 수 있다. 식(36)로부터 방식3에 의해 얻어지는 SNR_3 을 다음과 같다.

$$SNR_3 \equiv \frac{E[r^2(m, n)]}{\sigma_3^2} = p^2 \frac{E[r^2(m, n)]}{\sigma_{EI}^2} \quad (37)$$

유사하게 방식1과 방식2에 대해서도 SNR 을 정의할 수 있다.

$$SNR_i \equiv \frac{E[r^2(m, n)]}{\sigma_i^2}, \quad \text{where } \sigma_i^2 = E[e_i^2(m, n)], i = 1, 2 \quad (38)$$

식(37)과 식(38)로부터 SNR_3 을 SNR_1 과 SNR_2 로 표현하면 다음과 같다.

$$SNR_3 = \frac{p^2 \sigma_1^2}{\sigma_{EI}^2} SNR_1 = \frac{p^2 \sigma_2^2}{\sigma_{EI}^2} SNR_2 \quad (39)$$

식(39)은 $\sigma_{EI}^2 < p^2 \sigma_1^2$ ($\sigma_{EI}^2 < p^2 \sigma_2^2$)이면 방식1(방식2)에 비해 방식3이 우수한 복원 화질을 가짐을 보여준다. 여기서 비록 $\sigma_{EI}^2 > \sigma_1^2$ ($\sigma_{EI}^2 > \sigma_2^2$)이더라도 $p \geq 1$ 일 때 SNR_3 가 SNR_1 (SNR_2)보다 클 수 있다. 즉, EI 방식을 사용함으로써 치역 블록의 크기가 커지므로 σ_{EI}^2 가 증가할 수 있지만 σ_{EI}^2 가 $p^2 \sigma_i^2$ ($p^2 \sigma_2^2$)를 초과하지 않는다면 방식3은 방식1(방식2)에 비해 좋은 성능을 얻을 수 있다. 그래서 방식3은 방식1과 방식2에 비해 우수한 성능을 가지도록 하는 오차여유가 $(1-p^2)\sigma_1^2$, $(1-p^2)\sigma_2^2$ 만큼 각각 존재한다고 말할 수 있다.

방식3의 개선은 1차원 신호의 경우 명확하다. EI와 IEI는 1차원 신호의 경우 보간(interpolation)과 간축(decimation)과 같다. 치역 블록을 $r(n)$, 보간된 치역 블록을 $r_{EI}(n)$, 정합오차를 $e_{EI}(n)$ 라고 할 때, 그림7은 $p=2$ 인 경우 각 신호들의 스펙트럼을 보여준다. 그림 7(b)에서 보여지는 것처럼 간축을 위한 저대역 통과 필터링은 $r(n)$ 는 보존되면서 $e_{EI}(n)$ 의 고주파 성분이 제거되기 때문에 $IEI[e_{EI}(n)]$ 의 파워가 원래 신호 $r(n)$ 의 파워에 비해 상대적으로 작아짐을 그림 7(c)가 보여준다. 즉, 그림 7(a)에 나타난 $r(n)$ 의 파워는 그림 7(c)에서 보듯이 보존되지만 그림 7(b)의 $e_{EI}(n)$ 의 파워는 그림 7(c)에 p 배 만큼 감소한다. 그림 7에서 보듯이 고주파 영역에 높은 파워를 가진 오차신호에 대해

방식3은 SNR 측면에서 좀 더 유리하다. 방식3은 IEI의 과정에서 SNR 이득을 얻으므로 식(39)는 FILB방식에 대해서도 유효하다. 그래서 방식3과 FILB방식이 함께 적용될 때 그 성능은 좀 더 개선될 수 있다.

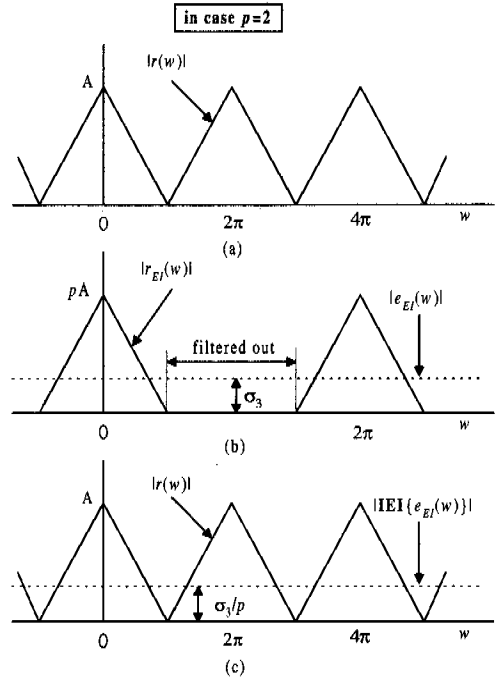


그림 7. 1차원 신호에 대한 EI의 효과

V. 실험 결과

경계 블록 형태 또는 그것의 정의역 블록 형태의 변형에 의한 3가지 방식이 각각 실험하여 비교되고 경계 블록에 대한 부호화를 마지막에 수행하는 방식인 FILB방식에 대해서도 실험적으로 성능을 평가한다. 편의상, FILB+방식1, FILB+방식2, FILB+방식3를 각각 FILB1, FILB2, FILB3로 표기하자. 실험에 사용된 영상은 352288 'Akiyo'와 'Mother and daughter'이고 영상들에 대한 마스크는 MPEG4에서 주어진 것이 사용되었다. 그림8은 배경 영역이 제거된 Akiyo 첫번째 영상과 MAD(mother and daughter) 두번째 영상을 보여준다. 성능 평가를 위해 계층적 분할은 수행하지 않았고 치역 블록과 정의역 블록의 크기는 8x8과 16x16이다. 정의역 블록을 찾는 간격 $\Delta_s=8$ 이고 방식2와 방식3에서 고려한

정의역 블록의 수는 치역 블록 주위의 64개이다. 그리고 서플 변환의 수는 8개이다. 그래서 하나의 치역 블록당 총 8x64개의 정의역 블록 후보들이 사용되었다. 방식1은 방식2와3과는 달리 임의의 모양을 가진 정의역 블록의 회전, 대칭 이동이 자유롭지 않기 때문에 4가지만 고려하는 대신 고려한 정의역 블록 후보들의 수는 128개로써 총 고려한 정의역 블록 후보들의 수가 방식2와 방식3과 같도록 하였다. 4가지 종류의 서플 변환만 고려한 것은 나머지 4개의 서플 변환이 임의의 모양에 대해서 유효하지 않기 때문이다. 방식1에서 고려한 4가지 종류의 서플 변환은 변환이 적용되지 않은 경우, 수직축과 수평축에 대한 대칭이동, 180° 회전이다. 예를 들어 치역 블록의 가로가 세로 보다 긴 경우 180° 회전은 유효한 반면 90° 회전은 유효하지 않다. 또한 밝기 값에 대한 축소 계수는 방식1, 방식2, 그리고 방식3에서 공통적으로 0.3, 0.4, ..., 1.0의 8가지를 고려하였다. 한편, 내부 블록들에 대해서는 경계 블록의 부호화를 위한 방식2와 방식3과 같은 조건을 주었다. 즉, 방식1, 2, 3 모두 8가지의 서플과 64개의 정의역 블록의 후보들이 고려되었다.



그림 8. 배경 영역이 제거된 Akiyo 첫번째 영상과 mother and daughter 두번째 영상

그림9는 Akiyo 영상의 경계 블록에 대한 부호화 결과를 보인다. 여기서 (a)는 경계 블록들의 PSNR이고 (b)는 내부 블록들의 PSNR이다. 방식1, 방식2, 방식3중 방식3가 가장 좋은 성능을 보임을 관찰할 수 있다. 이것은 제4장에서 분석한 것처럼 IEL과정에서 얻어지는 이득 때문이다. FILB를 사용한 방식인 FILB1, FILB 2, FILB3에서도 방식3을 사용한 FILB3가 가장 좋은 성능을 보임을 알 수 있다. 또한 FILB방식의 적용은 방식1과 방식3의 PSNR을 전체적으로 개선시켜줌을 관찰할 수 있다. 이는 식 (31)의 오차 성분 ϵ 가 0이 됨으로써 얻어지는 이득

이다. 그러나 방식2에 대한 FILB2의 개선 정도가 방식1과 방식3에 대한 FILB1과 FILB3의 개선 정도에 비해 적음을 관찰할 수 있다. 이것은 제4장에서 언급되었던 것처럼 방식2는 식(31)의 $||L_{21}||$ 를 감소시키는 효과에 의해 성능 개선이 이루어지므로 FILB와 방식2를 함께 사용하는 것은 FILB만을 적용한 경우와 비슷한 결과를 보이기 때문이다. 실제로 FILB만을 적용한 FILB1과 방식2, 그리고 FILB2는 유사한 결과를 보임을 관찰할 수 있다. 결과적으로 Akiyo영상에 있어서는 FILB3방식이 가장 좋은 성능을 보이고 방식1에 비해 약0.4dB의 이득을 가진다.

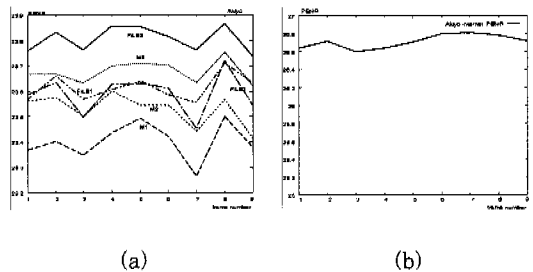


그림 9. Akiyo영상에 대한 제안된 방법들의 부호화 결과 (a) 복호화된 경계 블록들의 평균PSNR (b) 복호화된 내부 블록들의 평균 PSNR

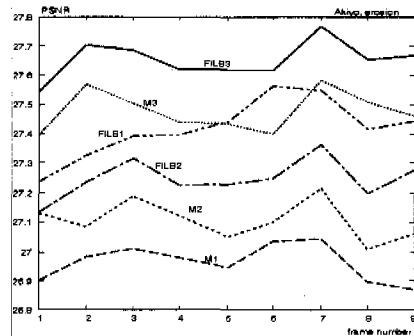


그림 10. 5x5 erosion 필터링에 의해 얻어진 새로운 마스크가 적용된 Akiyo 영상에 대한 제안된 방식들의 부호화 결과

그림 9(b)에서 보듯이 내부 블록의 PSNR이 경계 블록의 PSNR에 비해 상대적으로 우수한 것은 경계 블록은 배경 영역의 밝기 값과 물체 영역의 밝기 값을 모두 포함하므로 고주파 성분이 많기 때문이다. 그래서 경계 블록에 배경 영역이 포함되지 않도록 Akiyo 영상들의 마스크들에 대해 5x5 erosion

필터를 적용한 후 새로운 마스크들을 얻었다^[13]. 구조소자(structuring element) B에 의한 물체(object) X의 erosion은 B_x 가 X에 포함되도록 하는 모든 점들의 집합으로써 정의된다. 여기서 B_x 는 B의 원점(origin)이 x에 위치할 때 B의 변위를 나타낸다. 그림 10은 새로운 마스크들에 대해 적용된 제안된 방법들의 PSNR이다. 그림9와 마찬가지로 FILB3가 가장 좋은 성능을 보이고 방식1에 비해 약0.7dB의 이득을 가진다.

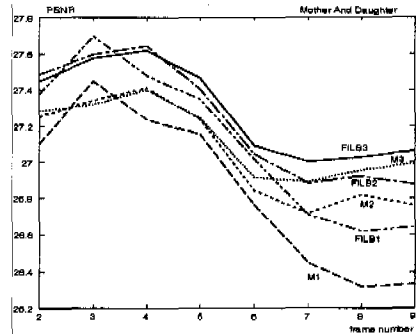
그림11은 MAD영상 2번부터 9번까지 적용한 결과이다. MAD의 첫번째 영상의 마스크는 2번부터 9번까지의 마스크와 상이하므로 2번부터 적용하였다. MAD 영상의 마스크의 크기는 Akiyo영상에 비해 작기 때문에 물체가 작은 경우이다. 그림11은 물체의 크기에 상관없이 유사한 결과를 보여준다. 즉, Akiyo영상에 대한 실험 결과와 유사하게 내부 블록의 부호화 결과가 경계 블록의 부호화 결과에 비해 우수하고 FILB3가 가장 우수한 성능을 보인다. FILB3는 방식1에 비해 최대 0.7dB정도의 개선을 보인다.

VI. 결론

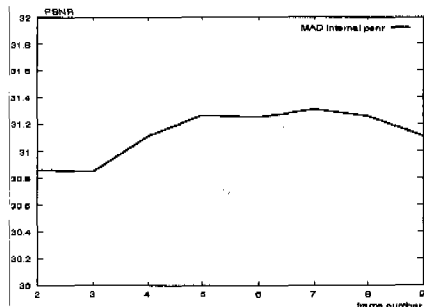
이 논문은 물체 기반 프랙탈 부호화 기법에 대해 기술하였다. 특별히 우리는 경계 블록들에 대한 부호화에 논의를 집중하였다. 제안된 물체 기반 프랙탈 부호화 기법은 널리 알려진 블록 기반 부호화 기법에 기초한다. 보통의 프랙탈 부호화와 비교하여 물체 기반 프랙탈 부호화는 경계 블록과 내부 블록이라는 두 종류의 지역 블록이 존재한다는 특징이 있다. 내부 블록에 대해서는 기존의 부호화 기법을 그대로 적용하였고 경계 블록들에 대해서는 블록의 형태 변형에 의한 3가지 방식과 블록의 부호화 순위를 이용한 FILB방식을 제안하였다. FILB방식은 경계 블록들의 정의역 블록이 내부 블록들로 구성되는 점에 착안하여 내부 블록을 먼저 부호화하고 복원한 뒤 경계 블록에 대한 부호화하는 방식이다. 방식1, 방식2, 방식3, 그리고 FILB방식의 성능 비교를 위하여 분석되고 실험되었다. 또한 우리는 방식 1, 방식2, 그리고 방식3의 성능은 FILB방식을 이용함으로써 좀 더 개선됨을 보였다. 결과적으로 경계 블록에 대해 FILB방식과 방식3을 같이 적용한 FILB3방식이 PSNR 측면에서 가장 우수한 성능을 보였다.

경계 블록은 전체 블록에 비해 소수이므로 제안된

방식들의 구현에 필요한 추가적 계산량은 무시될 수 있다. FILB방식의 구현은 부호화 순위를 조절함으로써 쉽게 구현되는 반면, 방식2와 방식3은 SADCT, ISADCT(inverse SADCT), IE, IEI, 등과 같은 추가적인 계산을 요구한다. 그러나 이 추가적인 계산은 중대하지 않다. 왜냐하면 프랙탈 부호화기의 정합 과정이 그들에 비해 매우 큰 비중을 차지할 뿐만 아니라 경계 블록의 수가 부호화될 전체 블록의 수에 비해 매우 작기 때문이다.



(a)



(b)

그림 11. Mother and daughter 영상에 대한 제안된 방식들의 부호화 결과 (a) 복원된 경계 블록들의 평균 PSNR (b) 복원된 내부 블록들의 평균 PSNR

참고 문헌

- [1] M. Barnsley, Fractals Everywhere, Academic Press, San diego, 1988.
- [2] A. E. Jacquin, "Fractal Image coding: A Review," Proceedings of the IEEE, Vol. 81, No. 10, Oct. 1993.
- [3] Y. Fisher, Ed., Fractal Image Compression:

- Theory and Application. New York: Springer-Verlag, 1995.
- [4] G. E. ien, S. Lepsy and T. A. Ramstad, "L2-optimal Attractor Image Coding With Fast decoder Convergence," Ph. D. Dissertation, The Norwegian Institute of Technology.
- [5] G. Caso, P. Obrador and C.-C. Jay Kuo, "Fast Methods For Fractal Image Encoding," Proceedings of VCIP'1995, SPIE Vol. 2501, part 1, 1995.
- [6] M. Kawamata, M. Nagahisa and T. Higuchi, "Multi-Resolution Tree Search For Iterated Transformation Theory-Based Coding," Proceedings of ICIP'94, Vol. 3, pp. 137-141, 1994.
- [7] G. Gaso and C.-C. Jay Kuo, "New Results For Fractal/Wavelet Image Compression," Proceedings of VICP'96, SPIE Vol. 2727, part 2, 1996.
- [8] O. C. Au, M. L. Liou and L. K. Ma, "Fast Fractal Encoding In Frequency Domain," Proceedings of ICIP'97, Vol. 2, pp. 298-301, 1997.
- [9] D. M. Monro and F. Dudbridge, "Fractal Approximation Of Image Blocks," Proceedings of ICASSP'92, Vol. 3, pp. 485-488, 1992.
- [10] H. Li, M. Novak and R. Forchheimer, "Fractal Based Image Sequence Compression Scheme," Optical Engineering, Vol. 32, No. 5, pp. 1588-1595, 1993.
- [11] S. Leopsy, G. E. Oien and T. A. Ramstad, "Attractor Image Compression With A Fast Non-Iterative Decoding Algorithm," Proceedings of ICASSP'93, Vol. 5, pp. 337-340, 1993.
- [12] H. S. Kang and S. D. Kim, "Fractal Decoding Algorithm For Fast Convergence," Optical Engineering, Vol. 35, No. 11, Nov. 1996.
- [13] A. K. Jain, "Fundamentals of digital image processing," Prentice Hall, 1989.
- [14] S. J. Cho, and S. D. Kim, "Texture Coding Using 2D-DCT Based on Extension/Interpolation (EI)," IEICE Trans. on Fundamentals of Electronics, Communications and computer Sciences, Vol. E90-A, No. 4, pp. 789-794, April 1997.
- [15] C. S. Park, D. D. Hwang, J. T. Lim, K. H. Chang, and S. D. Kim, "Daewoo Proposal for Region Texture Coding," ISO/IEC JTC1/SC29/WG11 MPEG95/m0555 Jan. 1996, Munich.
- [16] T. Sikora and B. Makai, "Shape-Adaptive DCT for Generic Coding of Video," IEEE Trans. Circuits & Syst. of for Video Technology, Vol. 5, No. 1, pp. 59-62, 1995.
- [17] M. Gilge, T. Engelhardt and R. Mehalan, "Coding of Arbitrary Shaped Image Segments Based on a Generalized Orthogonal Transform," Signal Processing: Image Communication, Vol. 1, No. 2, pp. 153-180, 1989.
- [18] S. F. Chang and D. G. Messerschmitt, "Transform Coding of Arbitrary-Shaped Image Segments," ACM Multimedia, Vol. 1, pp. 83-90, 1993.

강 현 수(Hyun-Soo Kang)

정회원



1969년 5월 18일생

1991년 2월 : 경북대학교 전자공학과 졸업

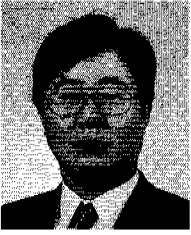
1994년 2월 : 한국과학기술원 전기및전자공학과 졸업 (공학석사)

1999년 2월 : 한국과학기술원 전기및전자공학과 졸업 (공학박사)

1999년 3월~현재 : 현대전자 정보통신연구소 선임 연구원

<주관심 분야> 영상처리, 영상부호화, 영상통신, 등

김 성 대(Seong-Dae Kim) 정회원



1953년 12월 26일생
1977년 2월 : 서울대학교 전자
공학과 졸업 (공학사)
1979년 2월 : 한국과학기술원
전기및전자공학과 졸업
(공학석사)
1983년 : 프랑스 INPT
ENSEEIHТ 졸업
(공학박사)

1984년~현재 : 한국과학기술원 전기및전자공학과 교
수

<주관심 분야> 영상처리, 영상통신, 컴퓨터 비전,
VLSI 구현 등

윤 병 주(Byung-Ju Yun) 정회원

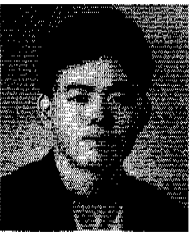


1970년 6월 30일생
1993년 2월 : 경북대학교 전자
공학과 졸업 (공학사)
1996년 2월 : 한국과학기술원
전기및전자공학과
졸업 (공학석사)

1996년 3월~현재 : 한국과학기술원 전기및전자공학
과 박사과정 재학

<주관심 분야> 영상부호화, 영상처리, 컴퓨터 비전
등

조 순 제(Soon-Jae Cho) 정회원



1971년 2월 15일생
1992년 2월 : 경북대학교 전자
공학과 졸업 (공학사)
1994년 2월 : 한국과학기술원
전기및전자공학과 졸업
(공학석사)
1999년 2월 : 한국과학기술원
전기및전자공학과 졸업
(공학박사)

1999년 3월 ~ 현재 : 삼성전자 재직중

<주관심 분야> 영상처리, VLSI 설계, 컴퓨터 비전
등