

## 빠른 수렴 속도를 가진 프랙탈 복호화 알고리즘

正會員 姜賢洙\*, 金聖大\*

## A Fractal Decoding Algorithm for Fast convergence

Hyun Soo Kang\*, Seong Dae Kim\* Regular Members

## 要 約

프랙탈 코딩은 부호화기의 계산량 감소와 성능 개선에 대해 주로 연구되어 왔다. 이는 프랙탈 코딩 기법이 만족할 만한 성능을 나타내기 위해서는 개선의 여지가 많음을 말한다. 그러나 부호화기의 개선 연구와 병행하여 이루어져야 할 부분인 복호화기에 대한 연구 또한 무시해서는 안된다. 현재 가장 널리 사용되고 있는 프랙탈 코딩 기법은 IFS(Iterated Function System)을 사용하기 때문에 반복변환 과정은 피할 수 없다. 이러한 반복변환 과정은 병렬 처리를 어렵게 하고 이에 따라 빠른 복호화기의 구현은 꼭 필요하다고 할 수 있다. 이 논문에서는 좀 더 수렴속도가 빠른 복호화기를 얻기 위한 방법을 제안하고자 한다. 실험 결과와 제안된 방법의 분석은 기존의 수렴속도보다 약 2배정도의 개선을 가져옴을 알 수 있었다.

## ABSTRACT

A fractal coding have been concentrated on releasing excessive computation burden and improving image quality in encoding. It is a reason that the solutions for them are not enough to apply to a coding system. But we don't have to ignore works for decoding. As the well-known fractal coding scheme is based on IFS(Iterated Function System), we cannot avoid the iterated transformation. Such transformation has difficulty in parallel processing and hence a fast decoding algorithm is necessary. This paper describes an algorithm to speed up decoding procedure. The analysis and the simulational results for the proposed method inform that our algorithm speeds up about 2 times in decoding.

\* 한국과학기술원 전기 및 전자공학과  
Department of Electrical Engineering Korea Advance  
Institute of Science and Technology  
論文番號 : 95097-0309  
接受日字 : 1995年 3月 9日

## 1. 서 론

프랙탈이란 모든 축척(scale)에서 자기 유사성(self-similarity)을 가지는 기하학적 구조를 의미한다. 이러한 자기 유사성에 기인하여 간단한 규칙에 의하여 복잡한 프랙탈 구조를 생성할 수 있다. 이것을 이용하여 공간 영역 상에 내포된 중복성(redundancy)을 제거함으로써 영상 데이터를 압축하는 것이 프랙탈 영상 압축 기법이다. 즉, 영상을 생성할 수 있는 규칙을 찾아냄으로써 그 규칙만으로 영상을 표현하여 압축 효과를 높이는 방법이다.

프랙탈이 영상에 널리 적용된 것은 최근 몇 년 사이의 일로 프랙탈의 여러 다른 특징을 이용한 여러가지 방법이 제안되었다. Barnsley<sup>(1)</sup>는 프랙탈의 자기 유사성을 이용한 IFS(Iterated Function System)을 기반으로 한 압축을 제안하였고, 그 이후로도 Jacquin<sup>(2)</sup>, Fisher<sup>(3)</sup>, Monro<sup>(4)</sup>, Novak<sup>(5)</sup>, 등과 같은 많은 사람들이 여기에 바탕을 둔 알고리즘을 제안하였다. 그러나 이러한 알고리즘들은 부호화기(encoder)부분의 성능 개선, 계산량의 감소에만 주력하였을 뿐, 복호화기(decoder)에 대한 연구는 소홀하였다. 이와는 다른 접근 방식으로 프랙탈 코딩을 내적공간에서 바라본 Oien과 Lepsoy는 빠른 부호화(encoding)과 복호화(decoding)에 대해 연구하였다<sup>(6,7)</sup>. 이들은 프랙탈에 사용된 변환의 각 항을 직교하는 기저(basis)로 구성하여 영상의 중복성(redundancy)을 제거하고 이에 따라 반복없는 복호화기의 구성을 가능하게 하였다. 이 알고리즘은 기저의 수를 많이 함으로서 변환 부호화(transform coding) 방식에 가까워지고 기저의 수가 적어짐으로서 수렴속도가 떨어진다. Jacquin의 방식은 Oien과 Lepsoy의 알고리즘의 가장 간단한 경우라고 할 수 있으므로 반복변환없는 복호화기의 구현이 어렵다. 본 논문에서 제안된 복호화 알고리즘은 Jacquin의 알고리즘을 기반으로 하고 그 때 수렴속도를 향상시킬 수 있는 방안을 제시한다.

이 논문의 구성은 프랙탈의 일반적인 알고리즘에 대해 설명하고 새로운 복호화기를 제안한 후, 이 복호화기에 대한 성능을 확률모델을 세워 분석한다. 또 이론적인 계산의 검정과정으로 실제적인 영상에 대한 실험을 수행하였다.

## 2. 이론적 배경

프랙탈 부호화에 있어서 우선적으로 만족되어야 할 조건은 축소성(contractivity)이다. 축소성에 의한 반복 변환은 고정된 하나의 영상을 만들어낸다. 이 영상이 전송하고자 하는 영상과 얼마나 가까운가가 프랙탈 부호화기의 성능을 좌우한다. 우선 프랙탈의 가장 중요한 성질인 축소성에 대해 살펴보자.

거리 공간(metric space)을  $(X, d)$ 라고 하자. 여기서  $d(\cdot, \cdot)$ 는 거리(metric)라고 한다.  $x, y \in X$  이고 변환  $W: X \rightarrow X$ 가 있을 때 다음 관계를 만족할 수 있다.

$$d(W(x), W(y)) \leq s d(x, y) \quad (1)$$

여기서  $0 \leq s < 1$  이면, 변환  $W$ 를 축소성(contractivity)을 가진다고 말한다. 변환  $W$ 가 축소성을 가지면 하나의 고정점  $x_f$ 를 가지게 된다.

$$W(x_f) = x_f \quad (2)$$

또 어떤  $x \in X$ 에 대하여  $\lim_{n \rightarrow \infty} W^n(x) = x_f$ 의 관계를 가진다. 그래서 임의의 점으로부터 변환  $W$ 를 반복적용하면 하나의 고정점으로 귀결된다. IFS은 이 축소 변환(contractive transform)들의 집합으로 이루어진다. 하나의 축소 변환  $W_i$ 로 이루어진 집합에 의해 유기되는 변환을 영상 공간(image space)  $(I(X), h)$ 에서 정의할 수 있다.

$$W(B) = \bigcup_{i=1}^N W_i(B), \quad \text{for all } B \in I(X) \quad (3)$$

이 변환 역시 하나의 고정점  $A \in I(X)$ 를 가지고 이를 attractor 라고 부른다.

$$A = W(A) = \lim_{n \rightarrow \infty} W^n(B) \quad (4)$$

여기서 일반적인 축소변환  $W$ 를 찾아내는 것이 쉬운 문제가 아니므로 프랙탈의 실용적인 응용에는 다음 식에 나타난 Collage theorem이 사용된다.

$$h(B, A) \leq (1-s)^{-1} h(B, W(B)) \quad (5)$$

식(5)는 주어진 영상( $B$ )과 attractor( $A$ )의 거리를 줄이는 문제를 주어진 영상과 collage  $W(B)$  사이의 거리를 줄이는 문제로 간편화함으로써 변환을 찾는 문제를 쉽게 만들어준다. 일반적으로  $W_i$ 는 affine trans-

form의 형태로 아래와 같이 표현된다.

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

여기서  $I(\cdot, \cdot)$ ,  $J(\cdot, \cdot)$ 는 지역블록과 정의역블록 내의 원소들의 화소 밝기값을 나타내고,  $a, b, c, d$ 는 geometric, shuffle transform을 의미하고  $s$ 는 축소성,  $(\delta x, \delta y, \delta g)$ '는 변위항(translational term)이다<sup>[23]</sup>. 여기서  $(\delta x, \delta y)$ 는 공간적 변위항(spatial translational term)으로 정의역블록의 위치를 결정해 주는 항으로서 영상 내에 어떤 블록을 정의역 블록으로 사용할 것인지를 결정해 주고  $\delta g$ 는 화소의 밝기값의 변위로서 정의역과 지역의 화소값 차를 표현해 준다. 특히  $\delta g$ 는 복원 영상의 화질에 직접적으로 관계된 항으로서 이에 대한 표현에 따라 여러가지 방식들이 제안되어져 왔다. 식(6)의 각 항을 찾아내는 과정을 부호화라고 하고 이 계수들을 이용하여 반복변환하는 과정을 복호화라고 한다.

### 3. 제안된 복호화 알고리즘

실제 영상에서 각 변환을 찾아내는 가장 일반적인 방법은 영상을 겹치지 않게 정해진 크기의 블록으로 나눈 뒤, 이를 지역블록으로 하여 Collage theorem에 따라 영상 자체 내에서 정의역블록을 찾아내는 것이다. 각 블록에 대해 찾아진 각 변환들을 이용하여 임의의 영상을 초기 영상으로 하여 반복변환하는 과정을 복호화라고 하는데 뒤에서 언급되는 '복호화'와 '변환'은 같은 의미로 사용된다.

앞 절에서 소개된 식(3)은 여러가지 변환의 합으로서 영상 전체에 대한 변환이 표현됨을 나타내고 있다. 각 변환  $W_i$ 은 하나의 지역블록에 대한 변환으로 지역블록은 정의역블록을 변환  $W_i$ 에 의해 생성된 결과적인 블록을 의미한다. 식(3)의 변환들은 모든  $i$ 에 대해 변환들이 동일한 영상  $B$ 에 적용되었다. 그러나 제안된 복호화 과정은 블록변환들  $W_1, W_2, \dots, W_N$ 가 순차적으로 적용됨에 따라 적용되는 영상을  $B, B^1, B^2, \dots, B^{N-1}$ 로 변화(update)시킴으로서 수렴속도를 증가시킨다. 여기서  $B^i$ 는 다음과 같이 정의된다.

$$B^i = \left[ \bigcup_{k=1}^{i-1} W_k(B^{k-1}) \right] \cup \left[ \bigcup_{k=i}^N P_k(B) \right],$$

여기서,  $B = \bigcup_{i=1}^N P_i(B)$ ,  $B^0 = B$  (7)

여기서  $B^0$ 는 블록변환이 한번도 수행되지 않은 영상 이므로 영상  $B$ 와 동일하고  $P_i(B)$ 는 영상  $B$ 를 겹치지 않도록  $N$ 개의 지역블록으로 나누었을 때  $i$ 번째 지역블록을 의미한다. 현재  $i$ 번째 블록을 복호화할 경우, 아직 변환이 수행되지 않은 부분  $\bigcup_{k=i}^N P_k(B)$ 를 영역  $M$ 이라고 하고 변환이 수행된 부분  $B^i - \bigcup_{k=i}^N P_k(B)$ 을  $M^c$ 이라고 하자. 이 때 영상  $B^i$ 는 복호화된  $i$ 개의 블록으로 이루어진 영역  $M$ 과 아직 복호화되지 않은  $N-i$ 개의 블록들로 이루어진 영역  $M^c$ 의 합으로 표현된다. 이렇게 현재까지 변환이 수행된 블록들을 다음 지역블록의 변환에 이용함으로써 수렴속도를 증가시킬 수 있다. 즉, 변환  $W_{i+1}$ 를 수행할 때 사용되는 영상  $B^i$ 는 이미 변환이 수행된 블록들의 영역  $M$ 를 포함하고 있으므로  $W_{i+1}$ 의 공간 변위항  $(\delta x, \delta y)$ 이  $M^c$ 영역에 위치하면 2번 이상의 변환을 수행한 효과를 가져온다. 그래서 새로이 정의된 변환  $W_{new}$ 는 다음과 같다.

$$W_{new}(B) = \bigcup_{i=1}^N W_i(B^{i-1}), \quad \text{for all } B \in I(X) \quad (8)$$

이렇게 변환을 구성함으로써 한번의 변환으로 여러 번의 변환을 수행한 결과를 얻을 수 있다. 이에 대한 설명이 그림2이다. 그림1은 기존의 방법으로서 변환하고자 하는 영상으로부터 변환을 블록별로 수행함을 나타낸다. 이는 각 블록들이 변환  $W_i$ 에 의해 한번의 변환을 수행한 결과를 가져온다. 반면에 그림2는 그림1과는 달리 초기영상의 변환이 수행됨에 따라 그 결과로부터 영상을 변화시키고 이 영상에 대해 다음 블록의 변환을 적용함을 나타낸다. 빗금친 부분이  $M^c$ 영역을 나타내고  $M$ 은 나머지 영역이다. 그림2에서 지역블록  $R_i$ 의 정의역블록이 이미 변환이 수행된 부분  $M^c$ 에 위치하고 정의역블록  $D_i$ 는 최소한 한번의 변환을 거친 화소들로 구성되어 있으므로 이를 정의역으로 사용하여 변환한  $R_i$ 는 최소한 두번의 변환을 거친 결과를 가져온다. 또 다른 경우로서 정의역이  $M$ 영역에 위치하는 경우이다.  $R_{i-1}$ 의 정의역  $D_{i-1}$ 은  $M$ 영역에 속하고 이는 한번의 변환도 가지지 않은 영역이므로  $R_{i-1}$ 은 한번의 변환을 거친 결과를 가져

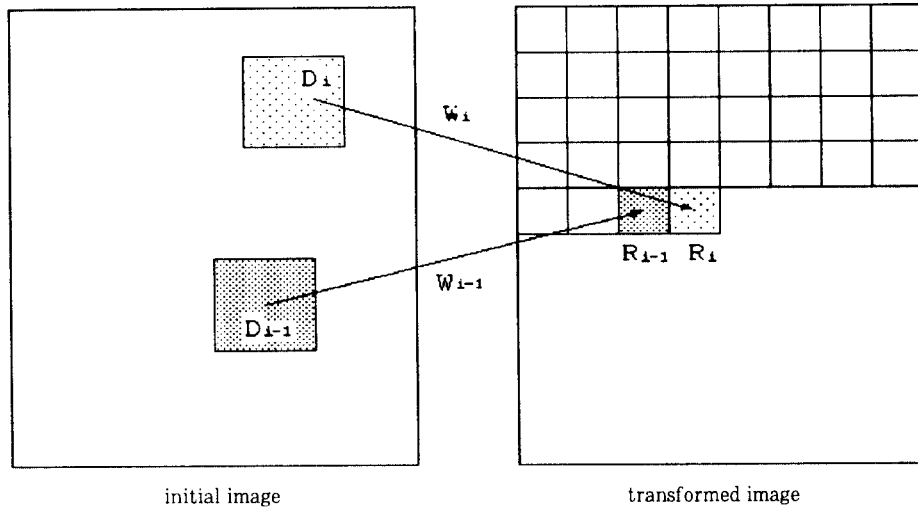


그림 1. 기존의 방법에 의한 복호화

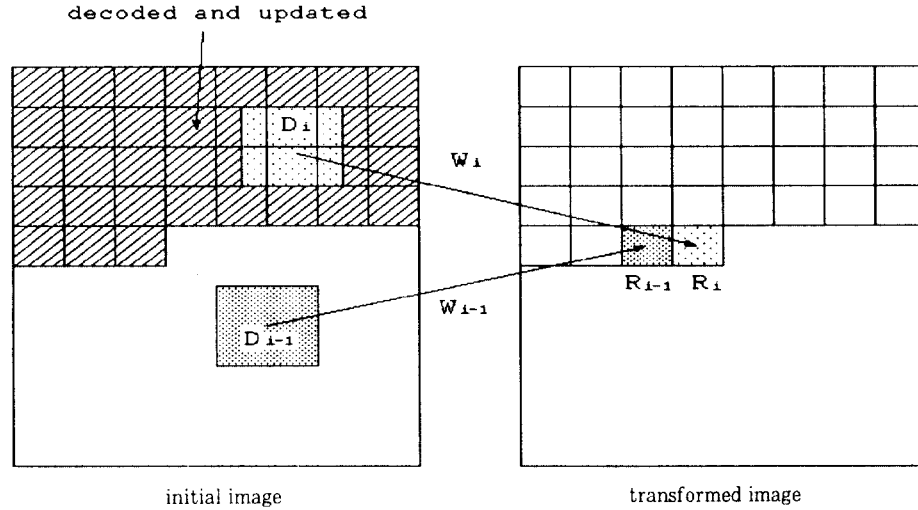


그림 2. 제안된 방법에 의한 복호화

온다. 그래서 전체 영상에 대해 평균적으로 볼 때 한번 이상의 변환을 거친 결과를 가져온다.

한편,  $i$ 가 증가함에 따라  $M$ 의 영역이 커지므로 정의역블록이 이미 복호화된 영역에 위치할 가능성이 많아지

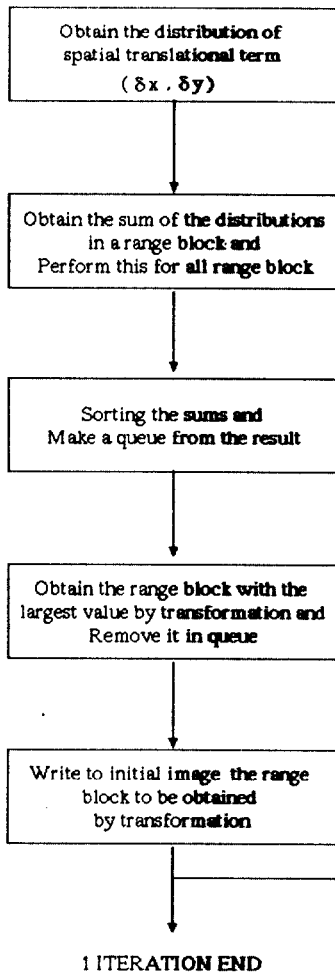
고 이로 인해 변환횟수의 증가를 가져올 것이다. 여기서 우리는 좀 더 생각해야 할 점이 있다. 정의역블록의 공간적 변위량이  $M$ 영역에 속할 확률을 높인다면 변환횟수의 증가를 가져올 수 있다는 점이다. 즉, 그림2에서

처럼 순차적으로 변환을 수행하는 것이 아니라 공간적 변위항의 확률 분포를 조사하여 가장 많이 선택되는 위치의 블록을 우선적으로 변환을 수행하는 것이다. 이는 그 다음 블록변환이 수행될 때 이미 변환이 수행된 부분을 정의역으로 사용할 가능성을 높이는 효과를 가진다. 이렇게 정의역블록의 분포를 이용할 경우 수렴속도를 극대화할 수 있다. 이는 다음 절에서 수학적으로 표현된다. 또 제안된 방법은 계산량의 증가가 없거나 약간의 증가로 수렴속도를 2배 정도 향상 시킬 수 있음을 주시해야 한다. 제안된 변환(복호화) 알고리즘의 흐름도가

그림3이다.

우선  $W$ 의 공간변위항  $(\delta x, \delta y)$ 의 분포를 조사한다. 이는 다음과 같은 과정으로 얻어진다.

1. 원영상 크기의 버퍼(buffer) 'dist'의 값을 모두 0으로 초기화한다.
2. 공간변위항  $(\delta x, \delta y)$ 는 정의역 블록의 맨 왼쪽 윗부분의 좌표이므로 치역블록의 정의역블록에 해당되는 영역  $(\delta x < x < \delta x + dx, \delta y < y < \delta y + dy)$ 의 버퍼값을 1만큼 증가시킨다. 여기서  $(\delta x, \delta y)$ 는 정의역블록의 크



```

/* INITIALIZATION */
for(y=0;y<height;y++)
  for(x=0;x<width;x++) dist[y][x] = 0;

/* THE DISTRIBUTION */
/* where (dx,dy) is domain block size */
while(all (delta x, delta y)){
  for(y=0;y<dy;y++)
    for(x=0;dx;x++) dist[y+delta y][x+delta x]++;
}

/* THE SUM OF DISTRIBUTIONS IN */
/* A RANGE BLOCK */
/* where (rx,ry) is range block size and */
/* generally dx = 2rx, dy=2ry */
for(y=0;y<height;y+=ry) for(x=0;x<width;x+=rx){
  count = 0;
  for(i=0;i<ry;i++) for(j=0;j<rx;j++)
    count += dist[y+i][x+j];
  TotalCountInRangeBlock[y/ry][x/rx] = count;
}

Sorting(TotalCountInRangeBlock);
while(for all range block) FractalTransform();
  
```

For all range block

그림 3. 제안된 복호화 알고리즘의 흐름도

기를 나타낸다.

3. 위의 과정을 모든 지역블록에 대해 수행한다.

위의 과정에 의해 얻어진 공간변위항의 분포는 하나의 화소에 대한 분포이고 우선적으로 변환할 블록을 결정하기 위해서는 블록단위의 분포를 얻어야 한다. 이를 위해서는 지역블록에 해당되는 영역의 분포들을 모두 합하여 그 값들을 비교하여 변환의 우선 순위를 결정한다. 이렇게 얻어진 순위에 따라 블록변환을 수행하고 변환에 의해 얻어진 지역블록을 초기영상에 써 넣는다. 이를 모든 블록에 대해 적용하면 1번의 변환이 끝난다. 이러한 변환 과정을 수렴된 영상이 얻어질 때까지 계속하면 된다.

이 절에서는 이렇게 정성적으로 언급이 되었지만 다음 절부터 이에 대해 확률 모델을 세우고 이를 풀어냄으로써 제안된 알고리즘의 변환횟수 증가 효과를 해석한다.

5.3. 제안된 방법의 평균 반복 수행 횟수의 계산( I )

이 절에서 계산하고자 하는 것은 복호화 시 임의의 초기 영상에서 변환  $W_{new}$ 를 한번 수행했을 때 제안된 알고리즘에서 얻을 수 있는 반복 횟수이다.

앞에서 이미 언급한 것과 같이 복호화 과정이 왼쪽에

서 오른쪽으로, 위에서 아래로 순차적으로 각 블록 또는 각 점(point)에 대해 이루어진다고 할 때, 이를 일렬로 나열했을 때 그림4처럼 나타낼 수 있다. 또 계산의 편의를 위해 블록들이 충분히 많아 연속적인 확률모델로 근사화가 가능하다는 가정으로부터 출발한다. 현재 복호화되고 있는 블록의 중심점 위치가  $a$ 라고 할 때, 또는 현재 복호화되고 있는 하나의 점  $a$ 를 생각할 때  $a$ 는 지역블록이고  $a$ 의 정의역블록의 위치  $x_2$ 를 생각할 수 있다. 이를 계속 확장하면  $x_n$ 은  $x_{n-1}$ 의 정의역블록이다. 예를 들어 정의역블록  $x_2$ 가  $a$ 보다 좌측에 있을 때, 즉, 이미 복호화된 부분일 경우와  $a$ 보다 우측에 있을 때, 즉, 아직 복호화되지 않은 부분일 경우로 나눌 수 있다. 정의역블록의 위치가 이미 복호화된 부분일 경우에는 정의역블록이 만들어지기 위해 취해진 반복변환횟수에 1을 더한 것이  $a$ 의 변환횟수이다. 변환이 한번 일어나는 경우는 그림4(a)처럼  $a$ 의 정의역블록 위치가 아직 복호화되지 않은 부분일 때이다. 또, 그림4(b)에서처럼 변환이 두번 일어나는 경우는  $a$ 의 정의역블록  $x_2$ 가 1번의 변환을 거친 경우이므로  $x_2$ 의 정의역블록  $x_3$ 가  $x_2$ 의 우측에 위치해야 한다. 그림4(c)는 3번의 변환을 거치는 경우이다. 이것을 일반화하여  $k$ 번의 변환을 거치는 경우를 생각할 수 있다. 여기서 정의역블록 위치  $x_n$ 의 분포는  $f(x_n)$ 으로 알려져 있고  $n$ 에 따른 각 분포는 서로 같고

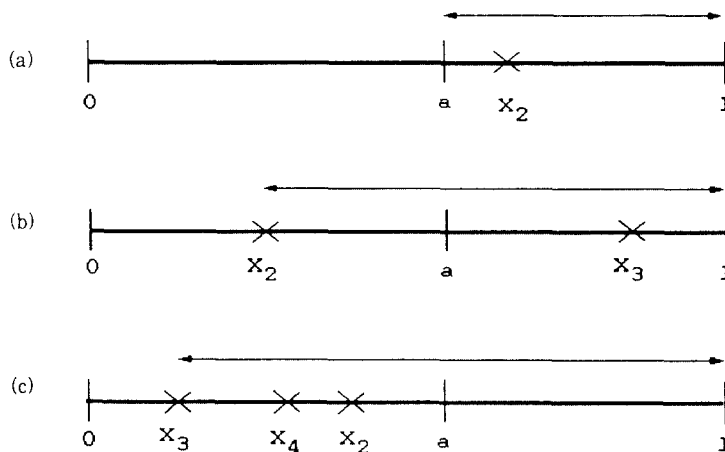


그림 4. 정의역의 위치에 따른 변환횟수  
 (a) 한번 변환이 일어나는 경우 (b) 두번 변환이 일어나는 경우  
 (c) 세번 변환이 일어나는 경우

독립(i.i.d)이라고 가정한다. 아래에서 사용된  $F(x)$ 는  $f(x)$ 의 확률축적함수(cumulative density function)이다.

- 변환이 1번 일어날 경우  

$$P\{n=1|a\} = P\{X_2 > a|a\} = \int_a^1 f(x) dx = F(1) - F(a) \quad (9)$$

- 변환이 2번 일어날 경우  

$$P\{n=2|a\} = P\{X_2 < a, X_3 > X_2|a\} = \int_0^a \int_{x_2}^1 f(x_2) f(x_3) dx_3 dx_2 = \int_0^a f(x_2) dx_2 - \int_0^a \int_0^{x_2} f(x_2) f(x_3) dx_3 dx_2 \quad (10)$$

- 변환이 3번 일어날 경우  

$$P\{n=3|a\} = P\{X_2 < a, X_3 < X_2, X_4 > X_3|a\} \quad (11)$$

- 변환이 k번 일어날 경우  

$$P\{n=k|a\} = P\{X_2 < a, X_3 < X_2, \dots, X_{k+1} > X_k|a\} = \int_0^a \int_0^{x_2} \dots \int_0^{x_{k-1}} f(x_2) f(x_3) \dots f(x_{k+1}) dx_{k+1} dx_k \dots dx_2 = \int_0^a \int_0^{x_2} \dots \int_0^{x_{k-1}} f(x_2) f(x_3) \dots f(x_{k+1}) dx_{k+1} dx_k \dots dx_2 - \int_0^a \int_0^{x_2} \dots \int_0^{x_{k-1}} f(x_2) f(x_3) \dots f(x_{k+1}) dx_{k+1} dx_k \dots dx_2 \quad (12)$$

- 변환횟수의 평균  

$$E\{n|a\} = \sum_{k=1}^{\infty} k \cdot P\{n=k|a\} = \sum_{k=1}^{\infty} k \cdot \left\{ \int_0^a \int_0^{x_2} \dots \int_0^{x_{k-1}} f(x_2) f(x_3) \dots f(x_k) dx_k dx_{k-1} \dots dx_2 - \int_0^a \int_0^{x_2} \dots \int_0^{x_{k-1}} f(x_2) f(x_3) \dots f(x_{k+1}) dx_{k+1} dx_k \dots dx_2 \right\} = 1 + \int_0^a f(x_2) dx_2 + \int_0^a \int_0^{x_2} f(x_2) f(x_3) dx_3 dx_2 + \dots + \int_0^a \int_0^{x_2} \dots \int_0^{x_{k-1}} f(x_2) f(x_3) \dots f(x_k) dx_k dx_{k-1} \dots dx_3 dx_2 + \dots = 1 + F(a) + \frac{F^2(a)}{2!} + \frac{F^3(a)}{3!} + \dots + \frac{F^k(a)}{k!} + \dots \quad (13)$$

$$= e^{F(a)} = \exp\left\{\int_0^a f(x) dx\right\}$$

계산된 결과에서 보듯이 복호화 위치가 a일 때 평균 변환횟수는  $e^{F(a)}$ 로서 a=0일 때 최소값 1을 가지고 a=1일 때 e로서 최대값을 가진다. a가 증가함에 따라 복호화된 부분이 많아지고 결과적으로 a의 정의역블록이 이미 복호화되었을 가능성이 크므로 변환횟수가 증가하는 효과를 가진다. 식(13)이 의미하는 또 다른 중요한 내용은 정의역블록의 확률 분포에 의존하여 반복횟수가 변화됨을 알 수 있다. 즉, 확률축적함수  $F(x)$ 가 빠르게 1에 접근함으로써 반복변환횟수를 증가시킬 수 있다. 이는 분포가 앞쪽으로 많이 몰려있을 경우에 해당하므로 정의역블록으로 많이 선택된 위치의 블록부터 복호화를 수행함으로써 보다 더 빠른 수렴속도를 얻을 수 있음을 말한다. 그림5은 균일 분포를 가진 경우와 지수 분포를 가진 경우의 반복 횟수의 비교결과이다. 여기서 x축은 치역블록을 labeling한 것으로 64개의 치역블록으로 영상이 구성되어 있다. 이 그림에서 볼 수 있듯이 균일 분포보다는 앞쪽에 분포가 밀집해 있는 지수 분포가 수렴 속도 면에서 훨씬 유리함을 관찰할 수 있다. 실제 영상의 경우 비균일 분포(non-uniform distribution)를 가지므로 이 분포를 이용하면 좀 더 빠른 수렴속도를 얻을 수 있음을 의미한다.

이제 영상 전체에 대한 평균 변환횟수를 알아보기로 하자. 식(13)은 치역블록이 a에 위치할 때 얻을 수 변환횟수를 의미하고 전 영상에 대한 평균 변환횟수를 얻기 위해서는 아래 식처럼 전체 구간의 평균을 취해야 한다.

$$E\{n\} = E\{E\{n|a\}\} = E\{e^{F(a)}\} = \frac{1}{l} \int_0^l e^{F(a)} da \quad (14)$$

계산의 편의를 위해 정의역블록의 위치분포가 균일 분포(uniform distribution)라고 가정했을 때 그 결과는 다음과 같다.

$$E\{n\} = \int_0^1 e^{\frac{a}{7}} \cdot \frac{1}{7} da = e - 1 \approx 1.72 \quad (15)$$

정의역블록의 위치 분포, 즉, 식(6)의 공간적 변위량 ( $\delta x, \delta y$ )의 분포가 균일 분포를 가질 경우, 첫번째 변환에서 1.72배의 변환 효과를 얻을 수 있음을 알 수 있다. 균일 분포는 제안된 알고리즘의 최악의 경우이므로 비균일 분포의 경우 더 큰 변환 효과를 얻을 수 있다.

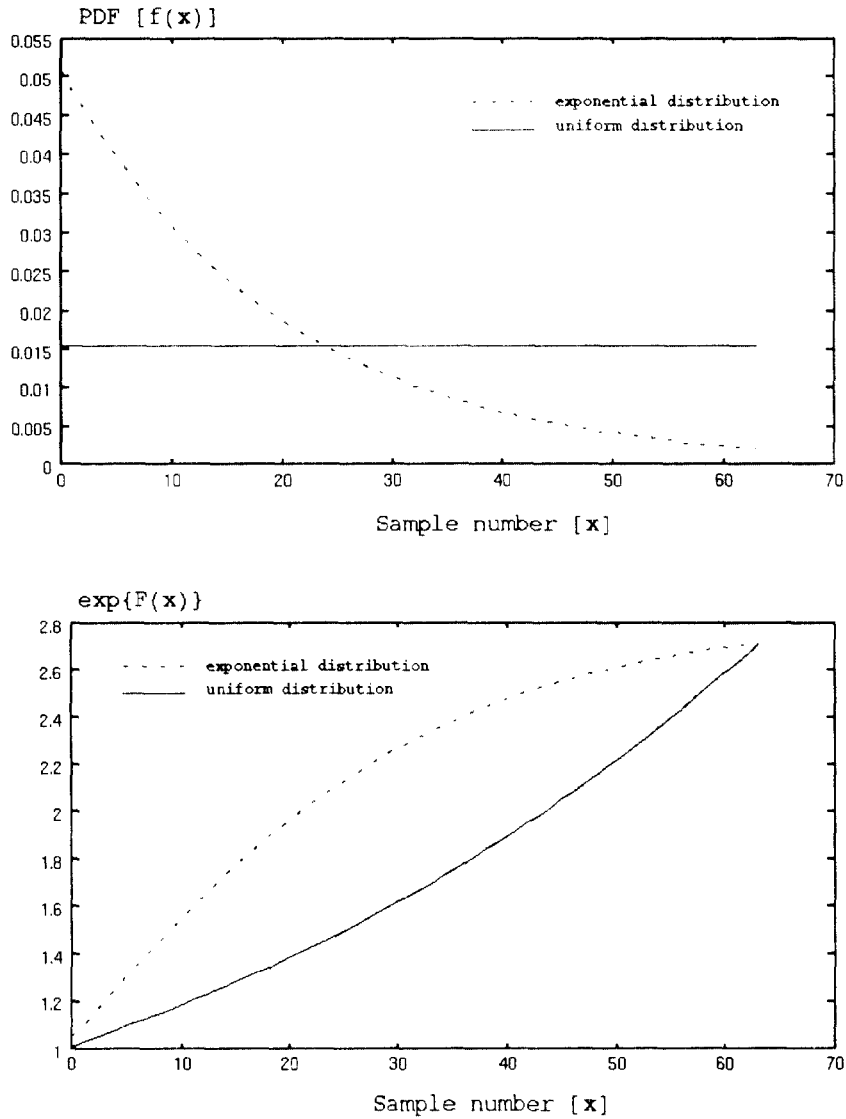


그림 5. 균일분포와 지수 분포의 반복변환횟수의 비교

이제까지는 초기 영상에서 첫번째 변환을 수행했을 경우  
에 대해 살펴보고 그 다음 변환들(두번째, 세번째, ...)에  
대한 변환 효과는 다음 내용에서 언급된다.

### 3.2. 제안된 방법의 평균 반복 수행 횟수의 계산(Ⅱ)

앞 절에서 소개된 방법의 계산과정은 다소 복잡한 면  
을 가진다. 그래서 이 절에서는 좀 더 편리하게 계산할  
수 있는 방법을 제시하고 이로부터 좀 더 일반적인 경우  
에 대해 전개된다.



3.2.1. 초기영상으로부터의 프랙탈 변환

초기영상에서 변환할 때, 변환횟수는 복호화하고 있는 위치와 다음과 같은 관계가 있다.

$$n(x, y) = \begin{cases} 1+n(x), & y > x \\ 1, & y \leq x \end{cases} \quad (16)$$

$$\begin{aligned} n(y) &\equiv E\{n(x, y)|y\} = \int_{-\infty}^{\infty} n(x, y)f(x|y)dx \\ &= \int_{-\infty}^{\infty} n(x, y)f(x)dx \end{aligned} \quad (17)$$

여기서  $y$ 는 현재 복호화하고자 하는 위치로서 앞 절에서 나타난  $a$ 에 해당되고,  $x$ 는  $y$ 에서의 변환 계수 중 공간적 변위, 즉,  $y$ 에 위치한 지역블록에 대한 정의역 블록의 위치를 의미한다. 식(16)은 현재 위치  $y$ 에서 복호화할 때 정의역블록의 위치  $x$ 가 이미 복호화 과정을 수행한 위치,  $y > x$ 라면  $x$ 에서의 반복변환횟수보다  $y$ 에서의 반복변환횟수가 1만큼 클 것이고 아니면 한번의 복호화를 수행한 것임을 의미하고 확률 변수(random variable)  $x, y$ 의 조합으로 표현되어 있다. 식(17)에서  $f(x|y)$ 는  $y$ 가 주어졌을 때  $x$ 의 확률밀도함수(probability density function)로서 복호화하고자 하는 지역블록의 위치  $y$ 는 정의역블록의 위치 분포에는 일반적으로 의존적이지 않으므로 독립으로 가정되었다. 여기서 우리가 관심을 가지는 것은 현재 복호화 위치  $y$ 가 주어졌을 때 평균 변환 횟수  $n(y)$ 이다. 이를 수식으로 표현하면 다음과 같다.

$$\begin{aligned} n(y) &= \int_{-\infty}^{\infty} n(x, y)f(x)dx \\ &= \int_0^y (1+n(x))f(x)dx + \int_y^l f(x)dx \\ &= 1 + \int_0^y n(x)f(x)dx \end{aligned} \quad (18)$$

식(18)을  $y$ 에 대해 미분함으로써 다음과 같은 형태로 바꾸어 쓸 수 있다.

$$\frac{dn(y)}{dy} = n(y)f(y), \quad \text{and } n(0) = 1 \quad (19)$$

이 때,

$$\frac{dn(y)}{dy} = f(y)dy,$$

$$\log\{n(y)\} = \int_0^y f(x)dx + C, C = \text{상수}$$

$$n(y) = C \exp\left\{\int_0^y f(x)dx\right\} \quad (20)$$

식(20)의 결과와 식(19)의 초기항을 이용하면 다음을 얻을 수 있다.

$$n(y) = \exp\left\{\int_0^y f(x)dx\right\} = e^{f(y)} \quad (21)$$

식(21)은  $y$ 에 따른 평균 반복횟수를 의미함으로 전체  $y$ 에 대한 평균은 다음과 같다.

$$E\{n(y)\} = \frac{1}{l} \int_0^l \exp\left\{\int_0^y f(x)dx\right\} dy \quad (22)$$

식(22)은 앞 절에서 계산한 식(14)와 같은 결과를 얻을 수 있다. 이렇게 복원된 영상을 다시 한번 제안된 방법으로 변환을 수행할 경우 몇번의 반복 효과가 있는지 다음 절에서 살펴보자.

3.2.2. 제안된 방법에 의한 평균 변환횟수의 일반화

우리는 앞 절에서 얻은 식(16)처럼 이 경우에도 마찬가지로 수식을 세울 수 있다. 그러나 차이점이 있다면 현재 복호화하고자 하는 위치  $y$ 가  $x$ 보다 작을 경우 1번의 변환이 아니라 이미 전체 영상에 대한 변환  $W_{new}$ 를 여러번 수행했으므로 이를 더해주어야 한다. 이전 반복 변환에서  $y < x$ 인 경우의 평균 변환횟수는 다음과 같이 얻어진다.

$$n_0(y) = \frac{1}{l-y} \int_y^l n_p(x) dx \quad (23)$$

여기서  $n_p(x)$ 는 이전 변환에서 얻어진 반복변환횟수이고  $n_0(y)$ 는 구간  $\{y, l\}$ 에서  $n_p(x)$ 의 평균 변환횟수로서 현재 변환횟수의 초기 값으로 사용된다. 예를 들어 두번째 변환일 경우 식(23)은 다음과 같다.

$$\begin{aligned} n_p(y) &= E\{n_p(x, y)|y < x\} \\ &= \frac{1}{l-y} \int_y^l \exp\left\{\int_0^x f(z)dz\right\} dx \end{aligned} \quad (24)$$

식(23)을 초기항으로 하는 변환횟수의 관계식은 다음과 같다.

$$n(x, y) = \begin{cases} 1+n(x), & y > x \\ 1+n_p(x), & y \leq x \end{cases} \quad (25)$$

식(25)를 이용하여  $y$ 가 주어졌을 때  $n(x, y)$ 에 대한 평균  $n(y)$ 는 다음과 같이 계산된다.

$$\begin{aligned} n(y) &= \int_{-\infty}^{\infty} n(x, y) f(x|y) dx = \int_{-\infty}^{\infty} n(x, y) f(x) dx \\ &= \int_0^y (1+n(x)) f(x) dx + \int_y^l (1+n_p(x)) f(x) dx \\ &= F(y) - F(0) + \int_0^y n(x) f(x) dx + F(l) \\ &\quad - F(y)(1+n_p(y)) = \int_0^y n(x) f(x) dx + l \\ &\quad + n_p(y)(1-F(y)) \end{aligned} \tag{26}$$

여기서,  $F(y) = \int_{-\infty}^y f(x) dx$

계산의 편의를 위해 두번째 변환이고 균일 분포를 가질 경우에 대해 식(26)을 계산해 보자. 이 경우 식(23)은 다음과 같다.

$$\begin{aligned} n_p(y) &= E(n_p(x, y) | y < x) \\ &= \frac{1}{l-y} \int_y^l e^{-x/l} dx = \frac{e^{-y/l} - e^{-l/l}}{1-y/l} \end{aligned} \tag{27}$$

식(25)를 이용하여 평균  $n(y)$ 를 얻을 수 있다.

$$\begin{aligned} n(y) &= \int_{-\infty}^{\infty} n(x, y) f(x|y) dx \\ &= \frac{1}{l} \int_0^y (1+n(x)) dx + \frac{1}{l} \int_y^l (1 + \frac{e^{-x/l}}{1-x/l}) dx \\ &= \frac{1}{l} \int_0^y n(x) dx - e^{-y/l} + 1 + e \end{aligned} \tag{28}$$

여기서,  $f(x|y) = \begin{cases} \frac{1}{l}, & 0 \leq x < l \\ 0, & \text{otherwise} \end{cases}$

이 식을 라플라스 변환(Laplace transform)을 이용하여 풀면 다음과 같은 결과를 얻는다.

$$n(y) = e^{1+y/l} - \frac{y}{l} e^{y/l} \tag{29}$$

위의 결과는  $y$ 에 따른 평균 반복횟수를 의미하므로 전체  $y$ 에 대한 평균은 다음과 같다.

$$\begin{aligned} E(n(y)) &= \frac{1}{l} \int_0^l n(y) dy \\ &= \int_0^l (e^{1+y/l} - \frac{y}{l} e^{y/l}) \frac{1}{l} dy \end{aligned} \tag{30}$$

이 결과에서는 제안된 알고리즘에 따라 2번 변환하면 평균적으로 3.67번을 반복한 것과 같은 효과를 지님을

말해준다. 위의 과정을 따라 제안된 알고리즘에 따라 3번 변환하면  $\frac{1}{2} e^2 + \frac{2}{3} e \approx 5.51$  번의 반복변환 효과를 가진다.

### 3.3. 불연속적인 확률모델로부터의 반복횟수의 계산

불연속적인 확률 변수  $x, y$ 에 대해서도 식(16)와 같은 관계가 성립한다.

$$n(x, y) = \begin{cases} 1+n(x), & y > x \\ 1, & y \leq x \end{cases} \tag{31}$$

이 식으로부터 모든 정의역블록의 위치에 대해 평균을 취하면 다음과 같은 결과를 얻는다.

$$\begin{aligned} E(n(x, y) | y) &\equiv n(y) = \sum_{x=0}^{N-1} n(x, y) P(X=x|y) \\ &= \sum_{x=0}^{y-1} (1+n(x)) \cdot P(X=x|y) + \sum_{x=y}^{N-1} 1 \cdot P(X=x|y) \\ &= \sum_{x=0}^{y-1} n(x) P(X=x|y) + 1 \end{aligned} \tag{32}$$

일반적으로 정의역블록의 위치분포(distribution of translational terms)는 치역블록의 위치와는 상관없으므로 불연속 확률 변수  $x, y$ 는 서로 독립이라고 가정한다.

$$n(y) = \sum_{x=0}^{y-1} n(x) P(X=x) + 1 \tag{33}$$

식(33)으로부터  $n(y-1)$ 은 다음과 같이 표현된다.

$$n(y-1) = \sum_{x=0}^{y-2} n(x) P(X=x) + 1 \tag{34}$$

식(33)과 식(34)로부터 다음을 얻을 수 있다.

$$n(y) = (1 + P(X=y-1)) n(y-1) \tag{35}$$

식(35)를 풀면 다음과 같은 결과를 얻는다.

$$n(y) = n(0) \cdot \prod_{x=0}^{y-1} (1 + P(X=x)) \tag{36}$$

그리고, 전체 영상에 대한 평균 변환횟수는 다음과 같다.

$$\begin{aligned} E(n(y)) &= \frac{1}{N} \sum_{y=1}^N n(y) \\ &= \frac{n(0)}{N} \sum_{y=1}^N \prod_{x=0}^{y-1} (1 + P(X=x)) \end{aligned} \tag{37}$$

초기영상으로부터 첫번째 변환을 수행할 경우  $n(0)=1$  이고, 정의역블록의 위치 정보가 균일한 분포를 가졌을 경우  $P(X=x) = \frac{1}{N}$  이다.  $\frac{1}{N}$ 을 식(35)에 대입하면 다음과 같이 주어진다.

$$n(y)=(1+1/N)^y \tag{38}$$

식(38)에서  $y=N-1$ 로 하고  $N$ 을 무한히 키우면 변환 횟수  $n$ 은  $e$ 에 접근하므로 앞 절에서 다루어진 연속적인 경우와 같은 결과를 나옴을 알 수 있다. 전체 영상에 대한 평균 변환횟수는 식(37)에 의해 다음과 같다.

$$E(n(y)) = (1+1/N)^{N-1} \tag{39}$$

여기서  $N \rightarrow \infty$ 이면  $(1+1/N)^N$ 은  $e$ 에 접근하므로  $N$ 이 커짐에 따라 연속적 확률 변수로 푼 결과와 같아진다.

#### 4. 실험 결과

실험에 사용된 영상은 512x512 lena이고 적용된 프랙탈 부호화 알고리즘은 Jacquini 제안한 방법을 사용하였다. Jacquini의 알고리즘은 지역블록을 2단계로 나누어, 변환이 정확하지 않은 지역블록에 대해서는 그 블록을 4개의 작은 블록으로 나누어 이들을 새로운 지역블록으로 간주하여 다시 프랙탈 변환 계수를 찾아내는 2단계 계층적 구조로 구성되어 있다. 그러나 본 논문에서는 확률 모델에 맞는 실험을 하기 위해 계층적 구조를 따르지 않고 1단계만 수행하도록 하였다. 사용된 지역블록의 크기는 8x8, 정의역블록은 16x16을 사용하여 512x512 크기의 영상 내에 지역블록이 4096개로 충분히 많이 존재하도록 하였다.

표1은 이미 추출된 프랙탈 변환들을 사용하여 복호화한 결과이다. 제안된 방법1은 공간변위항의 분포를 고려하지 않은 것이고 제안된 방법2는 분포를 이용하여 정렬 과정을 거친 결과이다. 복호화할 때 초기 영상은 모든 화소 값이 128로 일정하도록 하였다. 실험 영상을 프랙탈 부호화한 결과로서 얻어진 지역블록의 위치정보(translational term)의 분포는 그림6이다. 이 그림에서 보듯이 분포는 전체 영상 영역에 걸쳐 대체로 고르게 분포하지만 특정 부분에 집중되어 있는 부분이 관찰된다. 이러한 부분부터 복호화하는 것이 식(21)의 값이 최대가 되도록 한다. 이를 위해 많이 선택된 부분부터 정리하면 그림7과 같은 분포를 가진다. 많이 선택된 부분부터 복호화를 우선적으로 수행하고 이미 복호화된 부분(지역블록)을 그대로 이용한 결과가 표1에서 기존의 방법과 비교된다. 이 결과에서 보듯이 제안된 방법이 훨씬 빠른 수렴속도를 가짐을 알 수 있다. 기존의 방법으로 2번 변환했을 때 오차 성분의 표준편차 값과 제안된 방법으로 1번 변환했을 때 오히려 더 높은 PSNR을 얻을 수 있어 두 배 이상의 수렴속도를 가짐을 알려준다.

그림8와 그림9은 표1을 도식적으로 나타낸 것이다. 여기서 'proposed method not using distribution'이란 각 지역블록에 대해 영상의 윗쪽에서 아래쪽으로, 왼쪽에서 오른쪽으로 지역블록의 공간변위의 분포와는 상관없이 순차적으로 수행한 경우이다. 즉, 그 분포가 균일하다고 가정하고 제안된 방법을 적용했을 경우의 결과이다. 이 방법 역시 이미 변환된 지역블록을 이용하므로 위치정보의 분포를 이용한 경우보다는 성능이 떨어지지만 여전히 기존의 방법보다 훨씬 빠른 수렴속도를 얻을 수 있음을 알 수 있다. 위치분포를 사용하는 경우

표 1. PSNR과 표준편차(standard deviation)로 본 반복변환에 의한 수렴속도

반복 횟수	1	2	3	4	5	6	7	8
기존의 방법(PSNR)	18.77	22.99	26.90	29.62	30.62	30.75	30.76	30.77
제안된 방법1(PSNR)	20.36	26.86	30.18	30.72	30.77	30.77	30.77	30.77
제안된 방법2(PSNR)	23.34	28.81	30.57	30.77	30.77	30.77	30.77	30.77
기존의 방법(표준편차)	29.38	18.08	11.52	8.43	7.51	7.40	7.39	7.38
기존의 방법1(표준편차)	24.45	11.58	7.90	7.42	7.38	7.38	7.38	7.38
기존의 방법2(표준편차)	17.37	9.25	7.55	7.39	7.38	7.38	7.38	7.38

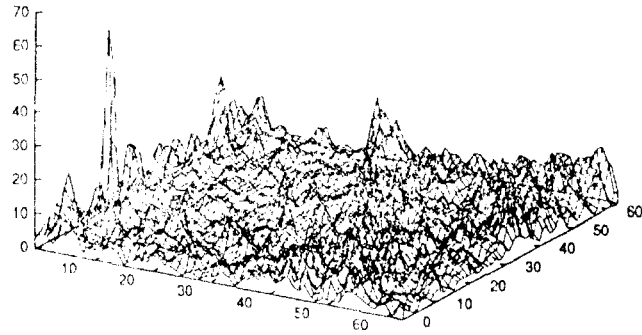


그림 6. 공간 변위항 ( $\delta_x, \delta_y$ )의 분포

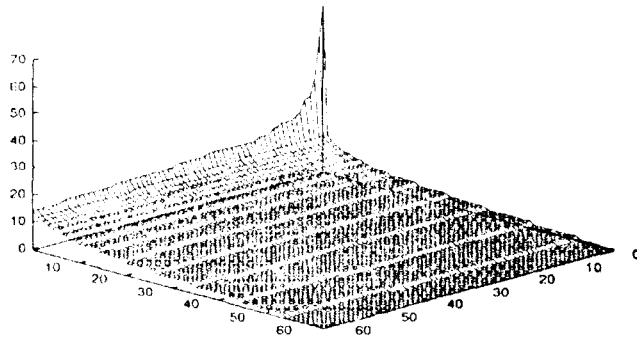


그림 7. 공간 변위항 ( $\delta_x, \delta_y$ )이 큰 부분부터 정렬(sorting)한 결과

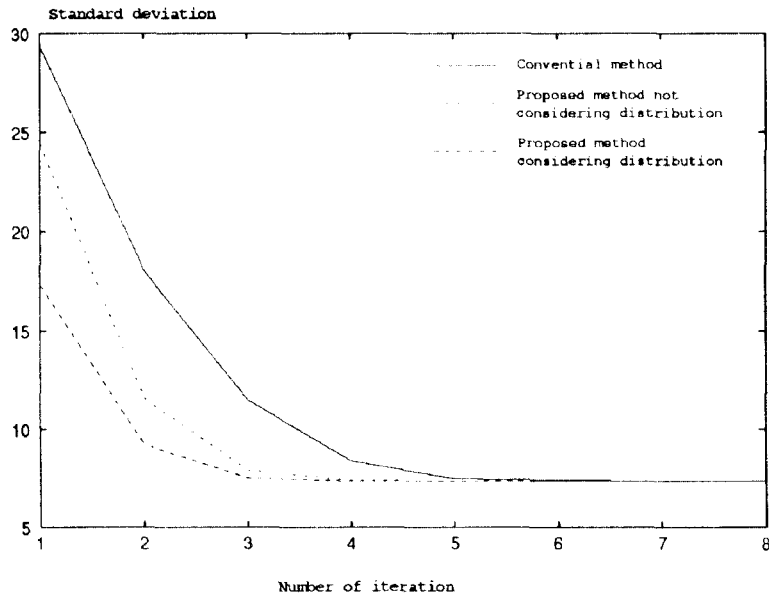


그림 8. 반복변환시 표준편차 변화의 비교

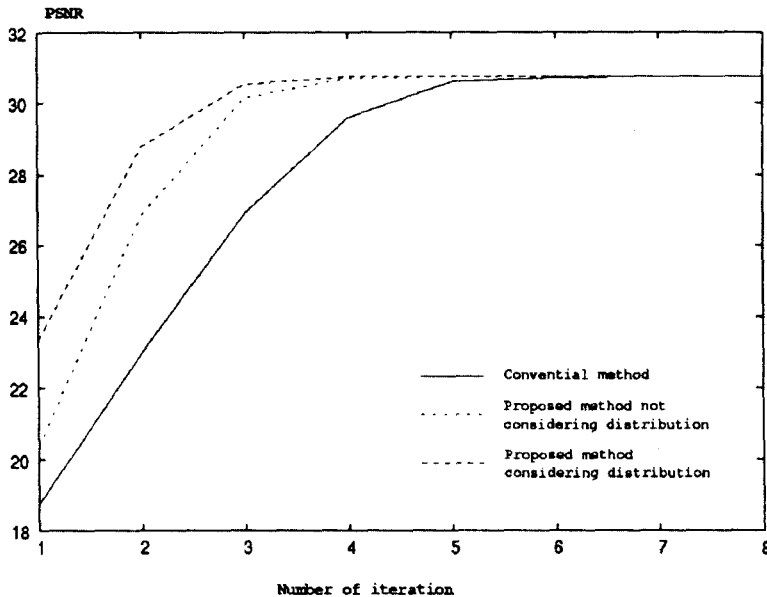


그림 9. 반복변환시 PSNR 변화의 비교

에는 그 분포들의 크기에 따라 정렬하는 과정이 더 첨가 되는 반면 위치분포를 사용하지 않는 경우는 정렬과정이 필요없지만 그 성능이 다소 떨어지는 단점을 가진다. 그림10은 반복변환시 PSNR 차와 표준편차의 차를 도식적으로 나타낸 것으로 제안된 방법이 빠르게 감소하여 수렴속도가 높음을 알 수 있다.

그림11과 그림12는 기존의 방법과 제안된 방법에 의해 두번의 변환과정을 거친 복원 영상이다. 그림11은 아직 치역블록들의 경계가 뚜렷하게 나타나 더 많은 반복변환이 필요함을 알 수 있고, 반면에 그림12는 부분적으로 화소값이 큰 부분이 존재하지만 많은 부분이 원 영상에 가까이 수렴했음을 알 수 있다.

실제 실험결과와 이론치가 어느 정도 차이를 보이는가를 살펴보기 위해 실험 영상에서 얻어진 분포를 이용하여 식(36)에 적용하여 이론치를 계산하였다. 그 결과가 그림13이다. 치역블록을 정렬된 차례로 변환해 나감으로서 변환횟수가 1에서 점점 증가함을 알 수 있다. 분포에 따라 균일한 분포를 갖는 경우보다는 실험 영상에서 얻어진 균일하지 않은 경우 좀 더 빠른 수렴속도를 얻을 수 있음을 알 수 있다. 이는 선택될 가능성이 많은 부분

을 우선적으로 변환을 해 놓음으로서 빠른 수렴속도를 얻을 수 있다. 그림13은 초기 영상에 한번의 변환을 수행했을 때 기존의 변환횟수보다 2배 이상의 효과를 얻을 수 있음을 말해준다. 그러나 실험에서 보듯이 실제 영상에서 얻어진 결과는 이론적인 결과보다 약간 뒤떨어진 결과를 얻었다. 이는 변환의 우선 순위를 블록 단위로 결정하기 때문이다. 이론적인 결과는 치역블록의 크기를 하나의 화소로 했을 경우이고 실제 실험에서는 그렇지 못하고 치역블록의 크기 만큼의 영역의 분포들을 합하여 우선 순위를 결정하였으므로 근소하지만 성능이 떨어진다.

그리고 제안된 방법에 의한 복원시 각 블록에 대한 변환횟수가 다를 수 있다. 이는 각 블록별 복원 화질이 다를 수 있음을 의미하기도 한다. 그러나 이는 문제가 되지 않는다. 디지털 영상 데이터의 경우 대략 8번 정도의 변환이면 거의 대부분 고정점에 다다르게 되어 더 이상의 반복 변환은 무의미하다. 예를 들자면, 축소성  $s$ 는 0과 1사이의 값을 가지므로 평균적으로 0.5의 값을 가진다고 하자. 그리고 최악의 경우 수렴해야 할 값과 초기 값과의 차가 255이다. 이 때  $n$ 번의 반복변환시 얻어지

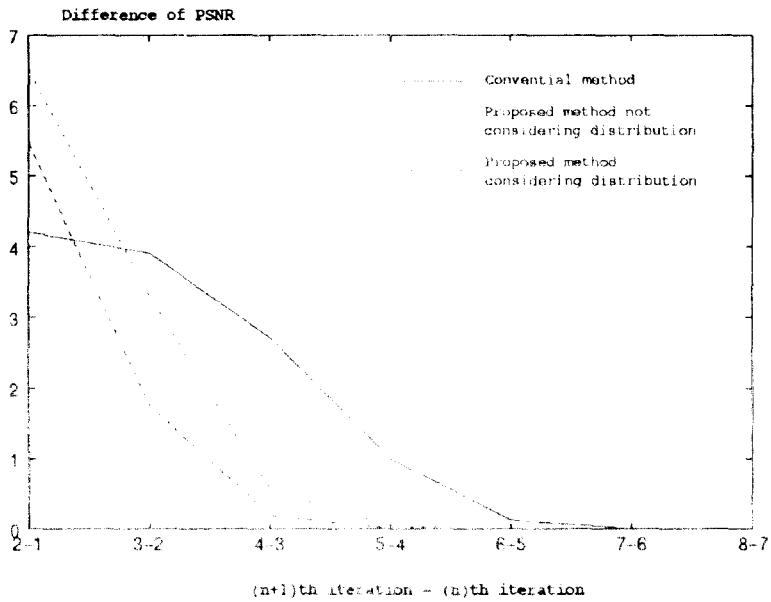
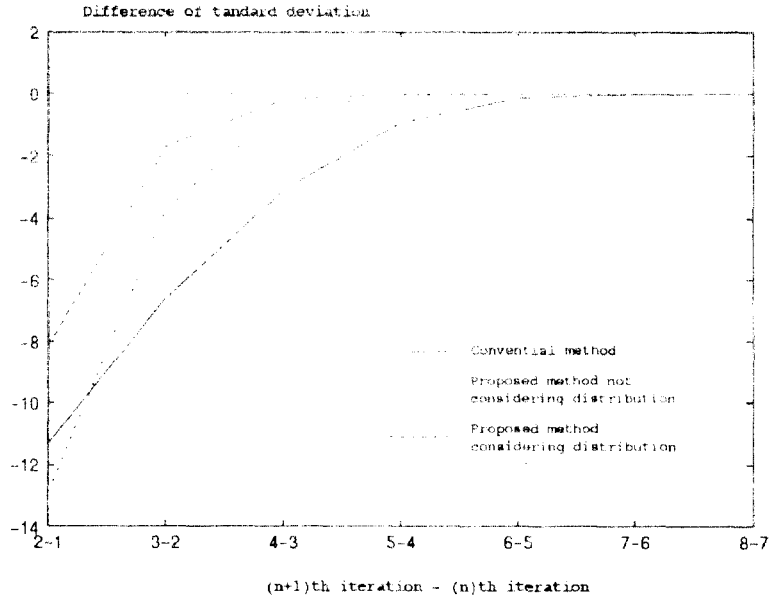


그림 10. 반복변환시 PSNR과 표준편차의 속도 변화의 비교



그림 11. 기존의 방법에 의한 두 번 반복변환 결과



그림 12. 제안된 방법(2)에 의한 두 번 반복변환 결과

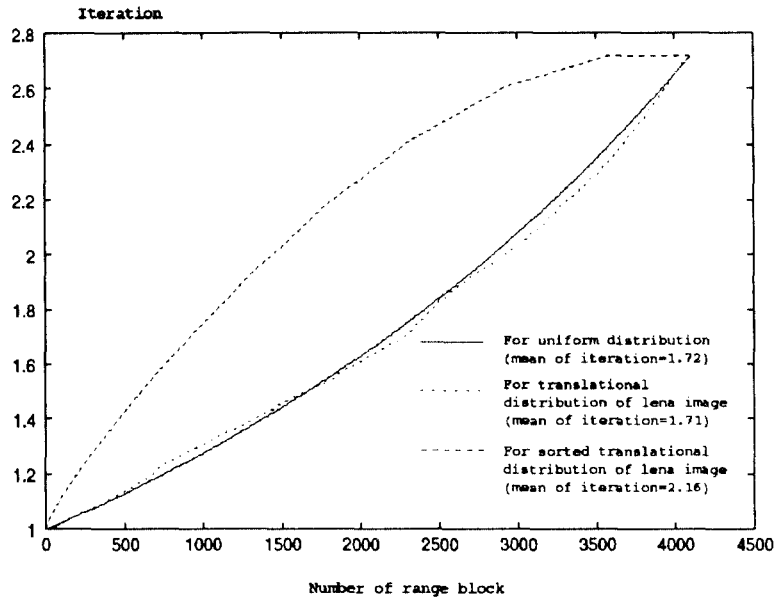


그림 13. lena 영상에 대한 반복변환횟수의 이론적 결과

는 오차 성분은  $0.5^n \times 255$ 이다. 이 때  $n=8$ 이면 이 오차 성분은 1이하가 되므로 그 이상의 변환은 의미가 없다. 이처럼 디지털의 경우 아나로그와는 달리 아주 많은 반복변환을 요구하지 않는다. 그래서 제안된 방법은 기존의 방법보다 약 2배의 수렴속도를 가지므로 4번 정도의 변환에 의해 대부분의 블록들이 고정점에 이르므로 블록 간의 변환횟수의 불균형에 의한 화질 저하는 없다고 말할 수 있다.

## 5 결 론

본 논문에서는 프랙탈 변환시 수렴속도를 증진시켜 좀 더 빠른 복호화를 수행할 수 있는 방안을 제시하였다. 프랙탈 변환시 이미 얻어진 결과와 변환 계수의 분포를 이용한 결과이다. 제안된 방법을 확률 모델을 세워 분석하고 이를 검증하였다. 본 문에서 보았듯이 영상에 대한 프랙탈 변환의 수렴속도는 기본적으로 지역블록의 축소성(contractivity)  $s$ 에 의존하지만 변환시 변환의 공간 변위항  $(\delta x, \delta y)$ 에 의존적임을 알 수 있다. 이로부터 우리는 프랙탈 부호화시 특별히 많이 선택되는 부분이 존재하도록 한다면 더 빠른 복호화 속도를 얻을 수 있음을

알 수 있다.

또한 제안된 알고리즘의 구현은 공간변위항의 분포에 따라 변환할 블록들의 순서를 정하는 부분이 첨가되어야 하므로 하드웨어적으로 좀 더 복잡해지긴 하지만 그만큼 빠른 수렴속도를 얻을 수 있다. 이러한 추가적인 하드웨어의 첨가의 부담을 없애기 위해서 정렬과정을 생략하고 순차적으로 변환을 수행한 것이 제안된 방법이고 실험결과와 'proposed method not using distribution'이다. 이 결과에서 보듯이 역시 빠른 수렴속도를 보임을 알 수 있다.

## 참고문헌

1. M. Barnsley, *Fractals Everywhere*. San Diego : Academic Press, 1988.
2. A. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations." *IEEE Trans. on Image Processing*, vol. 1, pp. 18-30, Jan. 1992.
3. Y. Fisher, "Fractal image compression." *SIG-GRAPH '92 Course Notes*, 1992.



4. D.M.Monro and F.Dudbridge, "Fractal approximation of image blocks," in Proc. ICASSP '92, vol.3, pp.485-488, 1992.
5. H.Li, M.Novak and R.Forchheimer, "Fractal-based image sequence compression scheme," Optical Eng., vol.32, no.7, pp.1588-1595, July, 1993.
6. G.E.Oien, S.Lepsoy and T.A.Ramstad, "An inner product space approach to image coding using contractive transformations," in Proc. ICASSP '91, pp.2773-2776, 1991.
7. S.Lepsoy, G.E.Oien and T.A.Ramstad, "Attractor image compression with a fast non-iterative decoding algorithm," in Proc. ICASSP '93, vol.5, pp.337-340, 1993.



姜賢洙(Hyun Soo Kang) 정회원

1969년 5월 18일생.  
 1987년 3월~1991년 2월 : 경북대학교 전자공학과 졸업(공학사).  
 1992년 3월~1994년 2월 : 한국과학기술원 전기및전자공학과 졸업(공학석사).

1994년 3월~현재 : 한국과학기술원 재학중.

\*주관심 분야 : 영상신호처리, 코딩.



金聖大(Seong Dae Kim) 정회원

1953년 12월 26일생.  
 1977년 : 서울대학교 전자공학과 졸업(공학사).  
 1979년 : 한국과학기술원 전기 및 전자공학과 졸업(공학석사).

1983년 : 프랑스 INPT ENSEEIHT 졸업(공학박사).

1984년~현재 : 한국과학기술원 전기 및 전자공학과 교수.

\*주관심 분야 : 영상처리, 영상통신, 컴퓨터 비전, VLSI 구현 등.