

# PC를 이용한 실시간 프랙탈 부호화 구현

正會員 金 載 喆\*, 朴 宗 植\*

## Real-time Fractal Coding Implementation Using the PC

Jae Chul Kim\*, Jong Sik Park\* *Regular Members*

※본 논문은 정보통신부에 시행된 대학기초연구지원사업비에 의하여 연구되었습니다.

### 요 약

연속적인 QCIF  $144 \times 176$  휘도 영상을 위한 실시간 프랙탈 부호기를 50MHz IBM 486 PC상에서 구현하였다. 프레임 부호화 속도와 데이터 압축율을 만족시키기 위하여 다음과 같이 알고리즘을 적용하였다. 부호화 시간을 단축하기 위하여 정의역 탐색이 없는 확장된 SAS(Self Affine System)을 이용하였다. 비트율을 줄이기 위해서 기존의  $4 \times 4$  치역블럭을  $8 \times 8$  블럭으로 확장하였고, 치역블럭의 확장으로 인한 복원 화질의 저하는 쿼드-트리 분할 방식을 적용하여 개선하였다.  $8 \times 8$  치역블럭 분할을 위하여  $8 \times 8$  치역블럭과 공간축소변환된  $8 \times 8$  정의역블럭의 유사성 정도에 따라 블럭 분할하여 블럭 분할로 인한 부호화시간을 최소화시켰다. 또한 치역블럭과 공간축소변환된 정의역블럭의 유사성 정도에 따라 프랙탈 계수의 개수를 변화시켜 화질 저하를 최소화하면서 전송 비트율과 부호화 시간을 단축시켜서 실시간 처리 가능하게 하였다. Claire 테스트 영상에 대하여 시뮬레이션한 결과, 평균 PSNR은 32.4dB, 비트율은 0.12 bpp 그리고 프레임당 부호화 시간은 33ms를 얻었다.

### ABSTRACT

Real time fractal coding for successive QCIF  $144 \times 176$  luminance images has been implemented on a 50MHz IBM 486 personal computer. To satisfy the frame encoding speed and data compression ratio, following algorithms are adopted. In order to minimize encoding time, extension SAS being not searching of domain blocks is used. For reducing the bits per pixel, conventional  $4 \times 4$  range block is extended to  $8 \times 8$  range block. And range block extension decrease quality of decoded image. For improvement quality of decoded image, the paper apply quad-tree partition method. In order to divide  $8 \times 8$  range block, self-similarity is compared  $8 \times 8$  range block with spatial contractive

\*경북대학교 전자공학과  
Dept. of Electronics, Kyungpook National University  
論文番號: 96220-0723  
接受日字: 1996年 7月 23日

transformed  $8 \times 8$  domain block. According to self-similarity, the block is partitioned and owing to block partition, increased encoding time is minimized. According to self-similarity of  $8 \times 8$  range block and spatial contractive transformed  $8 \times 8$  domain block, number of fractal factor is varied. Simultaneously with minimizing the decrement of decoded image's quality, transmission rate and encoding time is shorted. The results enable us to process the real-time fractal coding. For the Claire test image, the average PSNR was 32.4dB, 0.12 bit rates, and 33ms coding time per frame.

## I. 서 론

영상신호는 방대한 정보량을 갖고 있으며 효율적인 전송을 위하여 영상데이터의 압축은 필수적이다. 디지털 텔레비전에서 표준화된 안은 전송속도가 1Mbps 급인 MPEG-1과 2~20 Mbps급인 MPEG-2<sup>[1]-[3]</sup>등이 있으며 이들은 움직임의 예측과 보상, DCT, VLC 등 복합적인 압축 알고리즘을 이용하여 높은 압축율을 갖도록 하였다. 공중전화망이나 이동통신망과 같은 초저속 전송속도를 갖는 협대역 통신망을 이용한 음성 및 영상신호의 전송을 목적으로 매우 높은 압축율을 갖는 영상 부호화 연구가 진행중이며, 최근 프랙탈 이론을 이용한 영상 부호화에 대한 연구가 활발하게 이루어지고 있다.

프랙탈 이론<sup>[4]</sup>을 바탕으로 한 영상 부호화 기법은 자연계의 복잡하고 불규칙한 영상내에 존재하는 자기유사성을 이용하여 최적 축소변환 파라미터를 추출하여 이를 부호화하고 복호화시 임의의 초기영상에 축소변환을 반복적용시켜 원 영상을 복원한다<sup>[5]-[7]</sup>. 프랙탈 영상 압축 기법은 1990년 Jacquin<sup>[8]-[10]</sup>에 의해 처음 제안된 이래 Oien<sup>[11]-[13]</sup>, Fisher<sup>[14]</sup>, Monro<sup>[15]-[17]</sup> 등에 의해 많은 연구결과가 보고되고 있다. 프랙탈 영상 부호화기는 영상내에 존재하는 최적 변환연산자를 찾아야 함으로 계산량이 방대하여 실시간 처리기의 구현이 어렵다.

동영상 부호화를 위하여 Reusens<sup>[18]</sup>은 2차원의 지역블럭을 시간축 방향의 연속 프레임 지역블럭과 묶어서 지역 체적으로 나누고 3차원 탐색으로 프랙탈 부호화를 구현하였으나 방대한 계산량과 프레임 메모리가 소요된다. 매우 낮은 비트율(64kbps이하)을 가지는 부호화 방법을 제시한 Hurtgen<sup>[19]-[20]</sup>은 이전 프레임의 복원된 영상과 현재 프레임의 차 영상을 블럭 분할(segmentation)함으로써 부호화할 블럭을 선택하여 프레임내 부호화를 행하는 방법으로 Jacquin

의 부호화 방법을 수정하여 부호책(codebook)을 이용한다. 이 방법은 복원 영상을 얻기 위한 복원 시간, 부호책 탐색시간 그리고 부호화를 행할 블럭의 선택을 위한 블럭 분할이 부호화 시간에 포함 됨으로 전체 부호화 시간이 길어지는 단점이 있다. Wilson과 Monro<sup>[21]</sup>는 33MHz-i486 컴퓨터 상에서 탐색과정 없이 SAS (Self Affine System)모델을 이용하여 입력된 현재 프레임의 각 블럭과 이전 프레임의 대응하는 블럭을 비교하여 MAE(Mean Absolute Error)를 구하여, 그 오차가 큰 블럭 순서대로 부호화를 수행하고 전송 비트수를 피드백시켜 각 프레임에 부호화할 블럭 수를 조절함으로써 원하는 비트율을 얻을 수 있는 방법을 제안하였다.

본 논문에서는 50MHz-i486 컴퓨터상에서 프랙탈 영상압축 알고리즘을 이용하여 실시간 동영상 부호화기를 구현하였다. 시뮬레이션 결과, QCIF의 Claire 영상에 대하여 초당 평균 13.1 프레임, 움직임이 있는 블럭만을 선택하여 부호화하는 동영상에 대하여는 초당 30 프레임의 부호화 속도를 얻을 수 있었다.

PC상에서 전용하드웨어를 사용하지 않고 실시간 부호화를 이루기위하여 요구되는 프레임 전송율에 최대 구속조건을 정하여 계산량, 데이터 압축비 및 신호대 잡음비를 적합화 하였다. 이를 위하여 기존에 발표된 정지영상의 부호화시 적용되었던 알고리즘을 수정하여 결합하였으며 그 내용은 다음과 같다.

첫째, 정의역 탐색이 없는 확장된 타일링 구조를 구성하여 각 타일내에서 지역블럭으로 최소오차를 가지고 프랙탈 변환 식으로 부터 얻어지는 다원 1차 연립 방정식의 해를 구하여 프랙탈 부호화 계수를 얻었다.

둘째, 비트율의 감소를 위하여 쿼드-트리(quad-tree) 분할 방식<sup>[14]</sup>과 부호화 계수의 개수를 변화시켰다. 정의역블럭을 쿼드-트리 분할시 빠른 분할여부를 판정하기 위하여 지역블럭과 공간축소된 정의역블럭의 차이값을 이용하였다. 그리고 연속 영상에서 이전 프

레이프(frame)과 현 프레임 영상의 같은 위치 블록의 평균 자승 오차(Mean Square Error)를 구하여, 오차가 움직임 임계치 이하이면 부호화하지 않고 이전의 복원 영상을 이용하였다. 그리고 움직임이 있는 지역블록은 프레임내 부호화를 행한다. 프레임내 부호화는 지역블록과 공간축소된 정의역블록에 대한 평균 자승 오차가 평탄임계치 이하이면 2개의 계수, 이상이면 4개의 계수로 부호화를 하였다.

본 논문의 2장에서는 프랙탈 부호화 기법의 이론적인 배경에 대해 간단히 설명하고, 3장에서는 연속 영상을 위한 실시간 처리 알고리즘을 제안하였다. 그리고 4장에서는 제안된 방식의 성능을 비교, 분석하였고 5장에서 결론을 맺었다.

## II. 프랙탈 이론

### 1. IFS 이론

완전 척도 공간(complete metric space)  $(X, d)$ 에서 정의되는 변환  $W: X \rightarrow X$ 에 대해 다음의 식 (2-1)을 만족하는 상수  $0 \leq s < 1$ 가 존재하면, 변환  $W$ 를 축소 변환(contractive transformation)이라 한다<sup>[5]</sup>.

$$d(W(x), W(y)) \leq s \cdot d(x, y) \quad \forall x, y \in X \quad (2-1)$$

여기서  $d(x, y)$ 는 두 영상  $x, y$  사이의 거리를 나타내는 척도(metric)이고,  $s$ 는 변환  $W$ 에 대한 축소비(contractivity factor)를 의미한다. 이러한 유한개의 축소 변환으로 이루어진 계를 IFS(iterated function system)라 한다. 축소변환  $W$ 는 다음과 같이 축소선형영상변환  $Lx$ 와 영상신호  $x$ 에 무관한 영상  $t$ 로 분해할 수 있다.

$$W(x) = Lx + t \quad (2-2)$$

Collage 이론<sup>[5][22]</sup>에 의하면 끌개영상(attractor image)  $x_T$ 와 원 영상  $x_0$ 와의 거리는 다음의 관계를 만족시킨다.

$$d(x_0, x_T) \leq \frac{1}{1-s} d(x_0, W(x_0)) \quad (2-3)$$

식 (2-3)에서  $x_T$ 는 임의의 초기영상  $x_i$ 에 대하여 축소변환을 반복해서 수행하였을때 수렴하는 영상이다. 즉

$$x_T = \lim_{n \rightarrow \infty} W^n(x_i) \quad (2-4)$$

또한, 식 (2-2)를 반복 적용하면  $x_T$ 는 다음과 같이 표시된다.

$$x_T = \lim_{n \rightarrow \infty} \left( \sum_{i=0}^n L^{i-1} t + L^n x_i \right) \quad (2-5)$$

식 (2-5)의 우변의 우측항은  $\|L\| < 1$  이고,  $n$ 의값이 충분히 커지면 무시된다. 그러므로 끌개영상  $x_T$ 는 다음과 같이 표시된다.

$$x_T = \lim_{n \rightarrow \infty} \left( \sum_{i=0}^n L^{i-1} (t) \right) = (L^0 + L^1 + L^2 + \dots + L^\infty)t \quad (2-6) \\ = (I - L)^{-1} t$$

식 (2-6)에서  $I$ 는 단위행렬을  $(I - L)^{-1}$ 는  $(I - L)$ 의 역행렬을 의미한다.

원영상  $x_0$ 와  $W(x_0)$ 의 차영상을  $\delta$ 라 정의하면 다음과 같이 표시한다.

$$\delta = x_0 - W(x_0) \quad (2-7)$$

$$Lx_0 + t + \delta = x_0 \quad (2-8)$$

$$x_0 = x_T + (I - L)^{-1} \delta \quad (2-9)$$

따라서, 부호화시 원영상과 복호화된 끌개영상  $x_T$ 의 거리를 최소화 하기 위해  $\delta$ 를 최소로 하는 변환 매트릭스  $L$ 과  $t$ 를 얻어서 이들 정보를 부호화한다.

### 2. 타일링(Tiling) 구조

타일링 구조<sup>[17]</sup>는 다음과 같다. 원 영상을  $2B \times 2B$ 의 크기를 갖는 정의역블록  $D$ 로 분할하고, 분할된 블록을  $B \times B$  크기를 갖는 4개의 지역블록  $R$ 로 다시 분할한다. 그리고 정의역블록  $D$ 을 지역블록  $R$ 에 대응하도록  $2^2:1$ 의 비율로 공간축소변환을 하고 각 지역블록에 해당되는 축소변환  $W^{(k)}(D)$ 을 수행한다.

$$D = \sum_{i=0}^{2B-1} \sum_{j=0}^{2B-1} d(i, j) = \bigcup_{k=1}^4 R^{(k)} = \bigcup_{k=1}^4 \sum_{p=1}^{B-1} \sum_{q=0}^{B-1} r^{(k)}(p, q) \quad (2-10)$$

$$W(D) = \bigcup_{k=1}^4 W^{(k)}(D) = \bigcup_{k=1}^4 \hat{R}^{(k)} \quad (2-11)$$

그럼 1은 타일링 구조와 프랙탈 변환이 수행되는 과정을 표시한다.

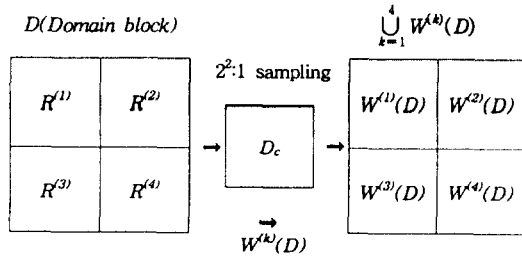


그림 1. 타일링 구조와 프랙탈 변환  
Fig 1. Tiling structure and fractal transform.

공간적 축소변환은 식 (2-12)에 표시된 샘플링(sampling) 방법<sup>[23]</sup>과 식 (2-13)과 같이 인근 화소값의 평균을 구하는 방법<sup>[10]</sup>이 널리 사용된다.

$$D_c = \sum_{i=0}^{B-1} \sum_{j=0}^{B-1} d_c(i, j) = \sum_{i=0}^{B-1} \sum_{j=0}^{B-1} d(2i, 2j) \quad (2-12)$$

$$D_c = \sum_{i=0}^{B-1} \sum_{j=0}^{B-1} d_c(i, j) = \sum_{i=0}^{B-1} \sum_{j=0}^{B-1} \left( \frac{d(2i-2j) + d(2i, 2j+1) + d(2i+1, 2j) + d(2i+1, 2j+1)}{4} \right) \quad (2-13)$$

시뮬레이션 결과 식 (2-12) 및 식 (2-13)을 이용하여 부호화된 영상을 복원하는 시뮬레이션을 다양한 영상에 대하여 수행한 결과 화질의 차이는 거의 없었다. 따라서 본 논문에서는 적은 계산량이 요구되는 식 (2-12)를 이용하여 부호화를 수행하였다.

### III. 동영상 부호화 알고리즘

동영상 부호화 방법인 MPEG-2<sup>[11]</sup>에서는 영상의 시간적 중복성의 제거로 압축율을 줄이기 위하여 움직임 추정(Motion Estimation)하고, 움직임이 추정된 영상으로부터 움직임이 보상된 영상과 원 영상의 차를 부호화한다. 일반적으로 차 영상에 대한 부호화 손실이 있는 프랙탈 부호화는 적용이 잘 되지 않는 것으로 알려져 있고, 실시간 프랙탈 동영상 부호화를 위하여 움직임의 추정은 연산량의 많다는 단점이 있다. 그러므로 프랙탈 동영상 부호화기는 움직임이 보상

된 영상을 이용한 차 영상에 대한 부호화의 어려움 때문에 프레임내 부호화가 주로 이용되며, 프레임내 부호화에서 복원영상의 화질은 정의역 탐색범위와 밀접한 관계를 갖는다<sup>[16]</sup>.

본 논문에서는 부호화의 소요 시간 陔 압축율에 관심을 두고 연산의 중복성을 줄여 실시간 처리 가능한 프랙탈 동영상 부호화 알고리즘을 제안한다. 그림 2는 제안된 연속 영상을 위한 프랙탈 부호화 알고리즘의 흐름도이다.

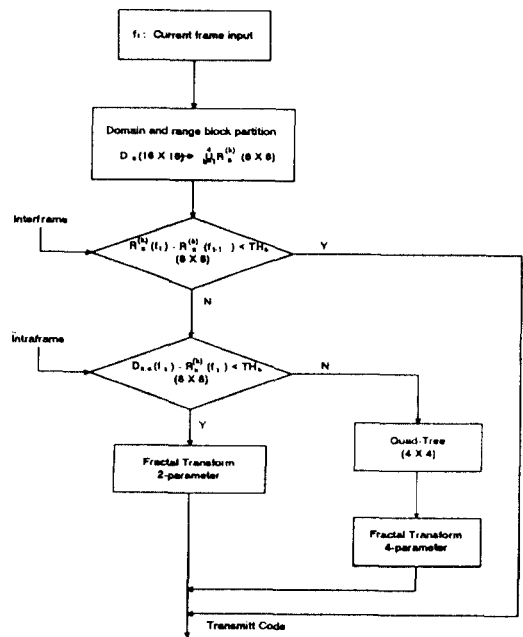


그림 2. 제안된 연속 영상을 위한 프랙탈 부호화 알고리즘의 흐름도

Fig 2. Flowchart of proposed fractal coding algorithm for successive images.

본 논문에서 제안된 동영상 부호화 과정은 입력된 현 프레임을 정의역으로 분할하고 정의역내의 지역 블럭들을 이전 프레임의 지역블럭과 비교하여 움직임이 발견되는 부분과 움직임이 없는 부분으로 나누어 이루어진다. 움직임이 없는 부분은 부호화를 수행하지 않고 이전의 영상을 이용하며, 움직임이 발견되는 부분은 프레임내 부호화를 한다. 프레임내 부호화에서 가장 큰 계산상의 부담인 정의역 탐색에 대한

시간을 최소화하기 위하여 탐색과정을 고정시킨 확장된 타일링구조를 이용한다. 부호화는 최소자승오차법<sup>[24]</sup>을 적용하여 얻어진 다윈 1차 연립 방정식의 해를 구하여 이루어진다. 본 논문에서는 D.L. Wilson 등<sup>[21]</sup>이 적용한 4×4 지역블럭을 비트율을 줄이기 위하여 지역블럭의 크기를 8×8로 증가시켰다. 지역블럭 크기의 증가로 인한 복원영상의 화질은 저하되고, 8×8 지역블럭을 쿼드-트리 분할 방식을 이용하여 4개의 4×4 지역블럭으로 분할하여 복원화질을 개선하도록한다.

블럭의 분할여부는 공간축소변환된 정의역블럭과 각각의 지역블럭들의 평균자승오차를 구하여 평탄임계치( $TH_S$ )와의 값을 비교하여 결정된다. 분할되지 않는 지역블럭은 2개의 변환계수로 부호화된다. 분할되는 지역블럭은 새로운 정의역블럭으로 바뀌고 4×4의 크기를 갖는 4개의 지역블럭으로 나뉘어져 4개의 변환계수를 사용하여 부호화한다.

자세한 부호화 과정을 설명하면 다음과 같다.

그림 2에서 연속적으로 입력되는 동영상은 현재 프레임( $f_i$ )과 이전 프레임( $f_{i-1}$ )을 일정한 크기의 16×16 정의역블럭( $D_n(f_i)$ ,  $D_n(f_{i-1})$ )들로 나누고 정의역내의 4개의 8×8 지역블럭( $R_n^{(k)}(f_i)$ ,  $R_n^{(k)}(f_{i-1})$ )들을 같은 위치 블럭에 대한 평균자승오차와 움직임임계치( $TH_m$ )를 비교하여, 그 오차가 작은 영역, 즉 움직임이 없는 영역은 부호화하지 않고 이전의 복원영상을 이용하여 비트율과 부호화 시간을 줄이도록 하였다.

$$IF((R_n^{(k)}(f_i) - R_n^{(k)}(f_{i-1}))^2 \leq TH_m) \quad (3-1)$$

$$\hat{R}_n^{(k)}(f_i) = \hat{R}_n^{(k)}(f_{i-1}) // \text{Nonmoving region}$$

$$IF((R_n^{(k)}(f_i) - R_n^{(k)}(f_{i-1}))^2 > TH_m) \quad (3-2)$$

// Moving region: intraframe coding

식 (3-1)과 식 (3-2)에서  $R_n^{(k)}(f_i)$ 은 현재 프레임의 n번째 정의역블럭의 k번째 지역블럭 영상,  $\hat{R}_n^{(k)}(f_i)$ 은 현재 프레임의 n번째 정의역블럭의 k번째 복원 지역블럭 영상,  $R_n^{(k)}(f_{i-1})$ 은 이전 프레임의 n번째 정의역블럭의 k번째 지역블럭 영상,  $\hat{R}_n^{(k)}(f_{i-1})$ 은 이전 프레임의 n번째 정의역블럭의 k번째 복원 지역블럭 영상을 나타낸다.

블럭의 분할여부는 다음과 같이 평탄임계치 ( $TH_S$ )

를 이용하여 결정된다.

$$IF((R_n^{(k)}(f_i) - D_{n,c}(f_i))^2 \leq TH_S) R_n^{(k)} = b_n^{(k)} + a_{3,n}^{(k)} D_{n,c} // \text{Nonpartition, Fractal coding } 8 \times 8 \text{ Range Block} \quad (3-3)$$

$$IF((R_n^{(k)}(f_i) - D_{n,c}(f_i))^2 > TH_S) // \text{Quad-tree Block Partition } 8 \times 8 \text{ Block to } 4 \times 4 \text{ Range Block} \quad (3-4)$$

$b_n^{(k)}$ 는 화소값 이동계수를  $a_{3,n}^{(k)}$ 는 화소값 비례 축소계수를 의미한다.

식 (3-3)은 정의역블럭과 지역블럭의 평균자승오차가 평탄임계치 ( $TH_S$ ) 보다 작을 경우 8×8 지역블럭에 대한 프랙탈 변환 식이다. 평균자승오차가 평탄임계치보다 클 경우는 4×4 지역블럭으로 쿼드-트리 분할을 한다.

식 (3-4)에서 분할된 영역에 대한 프랙탈 변환과정은 다음과 같다.

$$D_n^{(r)} = R_n^{(k)} = \bigcup_{m=1}^4 R_n^{(k,m)} \quad (3-5)$$

$$W_n^{(k)}(R_n^{(k)}) = \bigcup_{m=1}^4 W_n^{(k,m)}(R_n^{(k,m)}) \\ = \bigcup_{m=1}^4 (b_n^{(k,m)} + a_{1,n}^{(k,m)} x_i + a_{2,n}^{(k,m)} y_i + a_{3,n}^{(k,m)} D_{n,c}^{(r)}) \quad (3-6)$$

식 (3-5)에서는 8×8 지역블럭  $R_n^{(k)}$ 는 새로운 정의역블럭  $D_n^{(r)}$  된다. 이 정의역블럭은 4개의 지역블럭  $\bigcup_{m=1}^4 R_n^{(k,m)}$ 을 갖는다. 쿼드-트리 분할된 블럭의 공간 축소변환된 정의역블럭  $D_{n,c}^{(r)}$ 는 식 (3-7)과 같다.

$$D_{n,c}^{(r)} = \sum_{i=0}^3 \sum_{j=0}^3 d_{n,c}^{(r)}(i, j) \quad (3-7a)$$

$$R_n^{(k)} = \sum_{i=0}^7 \sum_{j=0}^7 r_n^{(k)}(i, j) \quad (3-7b)$$

$$D_n^{(r)} = \sum_{i=0}^7 \sum_{j=0}^7 d_n^{(r)}(i, j) \quad (3-7c)$$

$$\sum_{i=0}^3 \sum_{j=0}^3 d_{n,c}^{(r)}(i, j) = \sum_{i=0}^3 \sum_{j=0}^3 r_n^{(k)}(2i, 2j) = \sum_{i=0}^3 \sum_{j=0}^3 d_n^{(r)}(2i, 2j) \quad (3-8)$$

본 논문에서는 블럭의 크기에 따른 프랙탈 계수의 개수를 달리하여 부호화를 행한다. 8×8 지역블럭을 부호화하기 위하여 식 (3-9)을 최적화시키고 쿼드-트

리 분할된  $4 \times 4$  치역블럭을 부호화하기 위하여 식 (3-10)를 최적화시킨다.

$$d(R_n^{(k)}, W(R_n^{(k)})) = \sum_{i=0}^7 \sum_{j=0}^7 \{r_n^{(k)}(i, j) - [b_n^{(k)} + a_{3,n}^{(k)} \cdot d_{n,c}(i, j)]\}^2 \quad (3-9)$$

$$d(R_n^{(k,m)}, W_n^{(k,m)}(R_n^{(k,m)})) = \sum_{i=0}^3 \sum_{j=0}^3 \{r_n^{(k,m)}(i, j) - [b_n^{(k,m)} + a_{1,n}^{(k,m)} \cdot x_i + a_{2,n}^{(k,m)} \cdot y_j + a_{3,n}^{(k,m)} \cdot d_{n,c}^{(r)}(i, j)]\}^2 \quad (3-10)$$

식 (3-9)와 식 (3-10)에서 치역블럭에 대한 정의역블럭의 프랙탈변환은 원 영상과 오차  $d$ 를 최소화시키는 화소값 이동  $b_n^{(k)}$ , X축 비례 축소  $a_{1,n}^{(k,m)}$ , Y축 비례 축소  $a_{2,n}^{(k,m)}$ , 화소값 비례 축소  $a_{3,n}^{(k,m)}$ 를 구한다.

쿼드-트리 분할방식은 큰 블럭에 대하여 복잡한 계수를 적용하여 프레임당 분할된 블럭의 전체 계수를 줄임으로 비트율을 줄일수 있고, 큰 블럭의 변환으로 인한 화질의 저하를 블럭분할을 통하여 개선시킬 수 있다<sup>[23][25]</sup>. 이러한 방법은 원 영상과 복원영상과의 관계가 결정되어야 함으로 복원영상을 얻기위하여 복호화 시간이 부호화 시간에 포함이 되어야한다. 이는 실시간 부호화 측면에서 볼 때 부호화 시간에 부담이 크다. 그러므로 제안된 알고리즘에서는 치역블럭과 정의역블럭의 평균자승오차를 왜곡적으로 이용하여 블럭분할을 한다. 이러한 블럭분할을 위하여 복호화 과정을 부호화 과정에서 제거함으로써 실시간 부호화를 위한 부호화 시간을 최소화하였다. 그리고  $8 \times 8$  치역블럭을 식 (3-9) 과 식 (3-10)를 적용한 부호화 결과를 비교하면 화질은 거의 같고 전송 비트율 뿐만아니라 계산량 면에서도 식 (3-9)이 상당한 잇점이 있음을 시뮬레이션을 통하여 확인하였다.

프랙탈 부호화 계수를 구하기 위한 다원 1차 연립 방정식은 식 (3-11)와 식 (3-12)과 같다.

$$\begin{bmatrix} 64 & \sum_{i=0}^7 \sum_{j=0}^7 d_{n,c}(i, j) \\ \sum_{i=0}^7 \sum_{j=0}^7 d_{n,c}(i, j) & \sum_{i=0}^7 \sum_{j=0}^7 d_{n,c}(i, j)^2 \end{bmatrix} \begin{bmatrix} b_n^{(k)} \\ a_{3,n}^{(k)} \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^7 \sum_{j=0}^7 r_n^{(k)}(i, j) \\ \sum_{i=0}^7 \sum_{j=0}^7 r_n^{(k)}(i, j) d_{n,c}(i, j) \end{bmatrix} \quad (3-11)$$

$$\begin{bmatrix} 16 & 0 & 0 & \sum_{i=0}^3 \sum_{j=0}^3 d(i, j) \\ 0 & 80 & 0 & \sum_{i=0}^3 \sum_{j=0}^3 x_i d(i, j) \\ 0 & 0 & 80 & \sum_{i=0}^3 \sum_{j=0}^3 y_j d(i, j) \\ \sum_{i=0}^3 \sum_{j=0}^3 d(i, j) & \sum_{i=0}^3 \sum_{j=0}^3 x_i d(i, j) & \sum_{i=0}^3 \sum_{j=0}^3 y_j d(i, j) & \sum_{i=0}^3 \sum_{j=0}^3 d(i, j)^2 \end{bmatrix} \begin{bmatrix} b_n^{(k,m)} \\ a_{1,n}^{(k,m)} \\ a_{2,n}^{(k,m)} \\ a_{3,n}^{(k,m)} \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^3 \sum_{j=0}^3 r_n^{(k,m)}(i, j) \\ \sum_{i=0}^3 \sum_{j=0}^3 x_i r_n^{(k,m)}(i, j) \\ \sum_{i=0}^3 \sum_{j=0}^3 y_j r_n^{(k,m)}(i, j) \\ \sum_{i=0}^3 \sum_{j=0}^3 r_n^{(k,m)}(i, j) d_{n,c}^{(r)}(i, j) \end{bmatrix} \quad (3-12)$$

식 (3-10)에서  $b_n^{(k,m)}$ ,  $a_{1,n}^{(k,m)}$ ,  $a_{2,n}^{(k,m)}$ 는 식 (2-2)에서  $t$ 를 정의한다.  $x_i$ 와  $y_j$ 는 치역블럭의 중심을 기준으로 한 다음과 같은 화소 좌표값을 갖는다.

$$x_i = -3 + (i \times 2), \quad \text{for } 0 \leq i \leq 3 \quad (3-13a)$$

$$y_j = -3 + (j \times 2), \quad \text{for } 0 \leq j \leq 3 \quad (3-13b)$$

타일링 구조에서의 연립 방정식의 연산은 계산량을 상당히 줄여주는 잇점이 있다. 왜냐하면 (3-11)식에서 좌변의 계수는 정의역블럭의 화소값에 관한 계수이기 때문이다. 즉 정의역블럭에 포함된 4개의 치역블럭을 부호화할 때 정의역에 관계된 연산은 1번만으로 충분하기 때문이다.

본 논문에서는 부호화 시간을 최소화시키기 위하여 식 (3-11) 그리고 식 (3-12)의 해를 구하여 이용하였다.

식 (3-11)의 해는 다음과 같이 요약된다.

$$a_{3,n}^{(k)} = \frac{c_3 y_0 - c_0 y_3}{c_3^2 - c_0 c_6} \quad (3-14a)$$

$$b_n^{(k)} = \frac{y_0 - c_3 a_{3,n}^{(k)}}{c_0} \quad (3-14b)$$

$$c_0 = \sum_{i=0}^7 \sum_{j=0}^7 1 = 64 \quad (3-15a)$$

$$c_3 = \sum_{i=0}^7 \sum_{j=0}^7 d_{n,c}(i, j) \quad (3-15b)$$

$$c_6 = \sum_{i=0}^7 \sum_{j=0}^7 d_{n,c}(i, j)^2 \quad (3-15c)$$

$$y_0 = \sum_{i=0}^7 \sum_{j=0}^7 r_n^{(k)}(i, j) \quad (3-15d)$$

$$y_3 = \sum_{i=0}^7 \sum_{j=0}^7 r_n^{(k)}(i, j) d_{n,c}(i, j) \quad (3-15e)$$

식 (3-12)의 해는 다음과 같이 요약된다.

$$a_{3,n}^{(k,m)} = \frac{\left(y_3 - \frac{c_3}{c_0} y_0 - \frac{c_4}{c_1} y_1 - \frac{c_5}{c_2} y_2\right)}{\left(c_6 - \frac{c_3^2}{c_0} - \frac{c_4^2}{c_1} - \frac{c_5^2}{c_2}\right)} \quad (3-16a)$$

$$b_n^{(k,m)} = \frac{y_0 - c_3 a_{3,n}^{(k,m)}}{c_0} \quad (3-16b)$$

$$a_{1,n}^{(k,m)} = \frac{y_1 - c_4 a_{3,n}^{(k,m)}}{c_1} \quad (3-16c)$$

$$a_{2,n}^{(k,m)} = \frac{y_2 - c_5 a_{3,n}^{(k,m)}}{c_2} \quad (3-16d)$$

$$c_0 = \sum_{i=0}^3 \sum_{j=0}^3 1 = 16 \quad (3-17a)$$

$$c_1 = \sum_{i=0}^3 \sum_{j=0}^3 x_i^2 = 80 \quad (3-17b)$$

$$c_2 = \sum_{i=0}^3 \sum_{j=0}^3 y_j^2 = 80 \quad (3-17c)$$

$$c_3 = \sum_{i=0}^3 \sum_{j=0}^3 d_{n,c}^{(r)}(i, j) \quad (3-17d)$$

$$c_4 = \sum_{i=0}^3 \sum_{j=0}^3 x_i d_{n,c}^{(r)}(i, j) \quad (3-17e)$$

$$c_5 = \sum_{i=0}^3 \sum_{j=0}^3 y_i d_{n,c}^{(r)}(i, j) \quad (3-17f)$$

$$c_6 = \sum_{i=0}^3 \sum_{j=0}^3 d_{n,c}^{(r)}(i, j)^2 \quad (3-17g)$$

$$y_0 = \sum_{i=0}^3 \sum_{j=0}^3 r_n^{(k,m)}(i, j) \quad (3-17h)$$

$$y_1 = \sum_{i=0}^3 \sum_{j=0}^3 x_i r_n^{(k,m)}(i, j) \quad (3-17i)$$

$$y_2 = \sum_{i=0}^3 \sum_{j=0}^3 y_i r_n^{(k,m)}(i, j) \quad (3-17j)$$

$$y_3 = \sum_{i=0}^3 \sum_{j=0}^3 r_n^{(k,m)}(i, j) d_{n,c}^{(r)}(i, j) \quad (3-17k)$$

식 (3-11)의  $8 \times 8$  치역블럭에 대한 화소값 비례 축

소 계수  $a_{3,n}^{(k)}$  그리고 식 (3-12)의  $4 \times 4$  치역블럭에 대한 화소값 비례 축소 계수  $a_{1,n}^{(k,m)}$ 를 합하여  $a_3$ 로 나타낸다.

식 (3-11)과 식(3-12)의 좌변 행렬식의 값이 0을 가지면, 즉 다윈 1차 연립 방정식이 선형 독립이 아니며, 프랙탈 계수를 구하는 매트릭스의 연산은 불가능이 된다. 이러한 현상은 화소값의 변화가 거의 없는 평탄 블럭에서 많이 일어난다. 그리고 화소값 비례 축소 계수  $a_3$ 의 절대값이 1에 근접하게 되면 복호화시 수렴속도가 현저히 느려지고,  $a_3$ 의 절대값이 1보다 크면 발산이 가능하다. 그러므로 본 논문에서는  $a_3$ 의 절대값이 0.9 보다 클 경우는 0.5로 고정시켜서 실험을 하였다. 그림 3 및 그림 4는 CIF 288  $\times$  352 Alex 영상에 대한  $a_3$ 의 절대값을 고정시키지 않는 경우와  $a_3$ 를 0.5로 고정시켜 부호화를 수행한 Alex 영상의 복원을 위하여 프랙탈 변환을 7번 반복 적용하여 얻어진 복원영상을 보여준다. 그림 3에서 평탄영역에서 발산 또는 오실레이션 부분이 발생하는 것을 알 수 있다. 그림 4에서는 그림 3에서 발산 또는 오실레이션 부분이 사라진 것을 알 수 있다.



그림 3.  $|a_3| > 0.9$  경우,  $a_3$  값을 고정시키지 않을 때의 CIF Alex 영상의 복원영상

Fig 3. In case of  $|a_3| > 0.9$ , CIF Alex's decoded image for nonfixed scaling factor  $a_3$ .



그림 4.  $|a_3| > 0.9$  경우,  $a_3$  값을 0.5로 고정시킬때 CIF Alex 영상의 복원영상  
 Fig 4. In case of  $|a_3| > 0.9$ , CIF Alex's decoded image for fixed scaling factor  $a_3 = 0.5$

#### IV. 실험 및 결과

제안된 알고리즘의 프랙탈 동영상 부호화기의 성능을 추정하기 위하여 QCIF 포맷(144x176)의 Claire, Miss America 그리고 Alex 영상을 100 프레임 사용하여 50Mbps-1486 컴퓨터상에서 부호화를 수행하였다. 제안된 동영상 알고리즘의 평가를 위한 기준으로는 부호화시간, 화소당 비트수(BPP) 및 PSNR(peak-to-peak signal to noise ratio)을 적용하였다. 표 1은 시물레이션시 적용한 부호화에 소요되는 비트수를 각 계수에 대해 정리한 것이다. 표 2는 두 개의 계수를 갖는 분할되지 않는 8x8 지역블럭과 네개의 계수를 갖는 분할된 네개의 4x4 지역블럭으로 쿼드-트리된 8x8 블럭을 복화화시 소요되는 계산량에 대해서 정리한 것이다.

표 3에 100개 프레임의 Claire, Miss America 및 Alex 영상을 Monro의 알고리즘을 적용한 부호화 기법을 이용한 시물레이션 결과와 본 논문에서 제안된 알고리즘을 정지영상에 적용한 시물레이션 결과를 비교하였다. Monro의 알고리즘 적용시 각각의 영상에 대한 프레임당 평균 부호화 시간은 각각 97.90ms, 103.32ms, 116.05ms를 갖고 비트율은 1.25bpp로 대체로 높다. 각각의 테스트 영상에 대한 부호화 시간에

차이가 나는것은 식 (3-16 a)에서  $a_{3,n}^{(k,m)}$  를 구할때 크기가 0.9 보다 큰 블럭의 개수차이가 다른 것에 의한다. 본 논문의 시물레이션에서 각각의 100 프레임의 테스트에서  $a_{3,n}^{(k,m)}$  의 크기가 0.9 보다 큰 블럭은 총 블럭개수의 Claire 영상은 2.45%, Miss America 영상은 3.99%, 그리고 Alex 영상은 12.11%를 차지하였다.

표 1. 프랙탈 계수에 할당된 비트수  
 Table 1. Bits allocation for fractal parameters.

Motion Estimation	Classification Block	Luminant Shift	Gred X	Gred Y	Scaling
1	1	6	5	5	4

표 2. 8x8 블럭의 프랙탈 부호에 따른 연산량  
 Table 2. Calculation amount according to fractal coding of 8x8 block.

	Quad-tree partitioned four 4x4 range block								
	$ a_3  \leq 0.9$				$ a_3  > 0.9$				
	+	-	x	÷	+	-	x	÷	
long int	325	-	243	-	325	-	243	-	
float	5	22	14	25	197	86	206	38	
	Nonpartitioned 8x8 range block								
	long int	160	-	80	-	160	-	80	-
	float	-	3	5	2	64	67	69	3

본 논문에서 제안된 알고리즘을 평탄임계치  $TH_S = 64$ 로 설정하고 시물레이션 결과를 Monro의 부호화 기법을 이용한 시물레이션 결과를 비교하면 다음과 같다.

Claire 테스트 영상에 대한 프레임 평균화질은 0.14dB 떨어진 반면에 부호화 시간은 18.5%, 비트율은 50% 이상 줄일 수 있었다. 이는 기존의 쿼드-트리 방식<sup>23, 25, 26</sup>은 지역블럭과 복원된 지역블럭의 오차가 클 경우에 분할과 큰 블럭에서 프랙탈 계수의 개수를 증가 시킴으로 프레임당 블럭의 수를 줄여 비트율을 줄인다. 기존의 쿼드-트리 방식은 비트율에서는 잇점이 있으나, 부호화 시간이 부호화 시간에 포함되어 실시간 부호화기 구현을 어렵게하는 단점이 있다.

본 논문의 시물레이션은 정의역 탐색을 하지않는 확장된 SAS를 이용하기 때문에 부호화시간은 지역블럭의 분할과 프랙탈 계수의 개수에 밀접한 관계를 갖는다. 8x8 지역블럭의 분할을 위하여 지역블럭의 복원된 영상을 이용하지않고 8x8 지역블럭과 공간축소된 8x8 정의역블럭과의 유사성 정도에 따라 블럭 분할이 이루어진다. 그 결과, 8x8 지역블럭으로 부



표 3. 정지영상에 대한 제안된 알고리즘과 Monro의 부호화 기법과의 성능 평가.  
Table 3. Performance estimation of proposed algorithm and Monro's encoding method.

Image Block Partition		Claire			Miss America			Alex		
		PSNR	BPP	T(ms)	PSNR	BPP	T(ms)	PSNR	BPP	T(ms)
4 × 4 <sup>[21]</sup>		33.61	1.250	97.90	36.92	1.250	103.32	35.23	1.250	116.05
8 × 8, 4 × 4, Modified	TH <sub>s</sub> =64	33.47	0.610	79.74	35.50	0.705	86.58	34.77	0.978	106.93
	TH <sub>s</sub> =100	33.35	0.586	78.38	35.26	0.647	83.31	34.71	0.944	102.30
	TH <sub>s</sub> =225	33.09	0.544	75.55	34.53	0.517	76.09	34.02	0.860	104.30
	TH <sub>s</sub> =400	32.74	0.507	73.28	33.89	0.451	72.49	32.92	0.779	93.31

터 네 개의 4 × 4 치역블럭으로 분할 판정을 위해 소요되는 시간은 전체 부호화 시간을 증가시킨다. 따라서 Wilson<sup>[21]</sup> 등에 의한 알고리즘과 비교하여 볼 때, 8 × 8 치역블럭이 차지하는 비율이 높을수록 비트율 및 부호화 시간이 감소되는 효과를 갖는다.

표 4는 그림 2에서 제안된 동영상의 프레임간 중복성을 고려한 부호화 알고리즘의 성능에 대한 시뮬레

이션 결과를 나타낸다. 부호화할 블럭을 선택하기 위해 이전 프레임과 현재 프레임의 움직임 정도를 찾는 움직임 임계치와 현재 프레임내에서 정의역블럭과 치역블럭의 유사 정도를 나타내는 평탄임계치의 변화에 따른 부호화 성능 평가를 나타낸다.

본 논문의 시뮬레이션을 통하여 움직임이 있는 블럭중에서 TH<sub>s</sub> = 225이고 TH<sub>m</sub>이 5~15 사이로 변화

표 4. 제안된 알고리즘을 이용한 동영상에 대한 성능 평가.  
Table 4. Performance estimation of moving image using the proposed algorithm.

Image Threshold Value		Claire			Miss America			Alex		
		PSNR	BPP	T(ms)	PSNR	BPP	T(ms)	PSNR	BPP	T(ms)
5	TH <sub>s</sub> = 64	32.89	0.159	36.80	33.61	0.252	39.03	34.72	0.180	47.18
	TH <sub>s</sub> = 100	32.83	0.158	36.75	33.55	0.245	39.31	34.70	0.179	46.86
	TH <sub>s</sub> = 225	32.77	0.156	36.68	33.27	0.221	38.69	34.27	0.170	45.99
	TH <sub>s</sub> = 400	32.61	0.150	36.46	32.89	0.202	38.18	33.60	0.156	45.31
10	TH <sub>s</sub> = 64	32.48	0.122	32.87	32.30	0.191	32.46	34.25	0.114	40.37
	TH <sub>s</sub> = 100	32.44	0.121	32.86	32.27	0.187	32.12	34.23	0.113	40.22
	TH <sub>s</sub> = 225	32.40	0.120	32.78	32.12	0.172	31.93	33.88	0.107	39.72
	TH <sub>s</sub> = 400	32.31	0.116	32.60	31.86	0.160	31.58	33.39	0.098	39.28
15	TH <sub>s</sub> = 64	32.17	0.102	30.80	31.29	0.160	30.04	33.78	0.094	37.10
	TH <sub>s</sub> = 100	32.13	0.102	30.79	31.27	0.158	30.04	33.77	0.094	37.02
	TH <sub>s</sub> = 225	32.11	0.101	30.78	31.16	0.148	29.88	33.56	0.089	36.67
	TH <sub>s</sub> = 400	32.04	0.098	30.64	30.98	0.139	29.63	33.09	0.082	36.35

되면 Clarie 영상에서는 움직임이 있는 영역의 1.5%~2.8%, Miss America 영상에서는 12.3%~24.0%, 그리고 Alex 영상에서는 8.4%~9.8%가 8×8 치역블럭으로 판정되었다.

본 논문에서 제안된 알고리즘은 8×8 치역블럭을 기본으로 하므로 4×4 치역블럭을 기준으로 하는 Wilson<sup>[21]</sup>의 부호화 알고리즘과 비교할 때, 프랙탈 부호화 계수의 개수 이외에 치역블럭의 전송유무 또는 치역블럭 어드레스를 나타내는 비트가 상대적으로 줄어든다. 블럭분할여부에 할당된 비트수를 포함하여도 줄어든 비트수는 일정한 전송율에서 프랙탈 부호화 계수 비트의 비율을 증가시키는 잇점이 있었다.

제안된 알고리즘에서는 움직임 및 평탄임계치들의 값에 따라 복원 영상화질, 비트율 및 부호화 시간은 가변적이다. 표 4에서  $TH_m=10$ ,  $TH_S=225$  일 때, 평균 부호화 시간은 Claire, Miss America 영상에 대하여 32.78ms, 31.58ms 소요됨으로 초당 30 프레임, Alex 영상은 39.28ms로 초당 25 프레임의 속도로 실시간 부호화 처리가 가능하였다.

그림 5은 100프레임의 동영상을  $TH_S=255$ 로 전체 부호화를 행한 각 프레임에 대한 PSNR의 변화이다. 그림 6과 그림 7에서는  $TH_m=10$ ,  $TH_S=255$ 으로 정했을 때, 각 프레임에 대한 복원된 영상의 PSNR과 비트율의 변화를 나타낸다. 그림 8은 연속의 Clarie 테스트 영상 100번째 프레임에서 200번째 프레임 사이에서 많은 움직임이 있는 영상에 대한 복원영상이다.

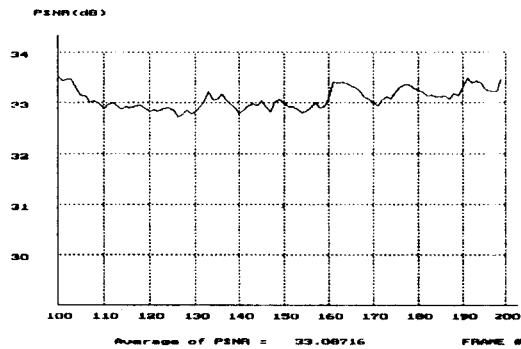


그림 5.  $TH_S=225$ , Claire 영상에 대한 정지영상 부호화에 따른 PSNR 변화.  
Fig 5.  $TH_S=225$ , PSNR variation according to still image encoding of Clarie images.

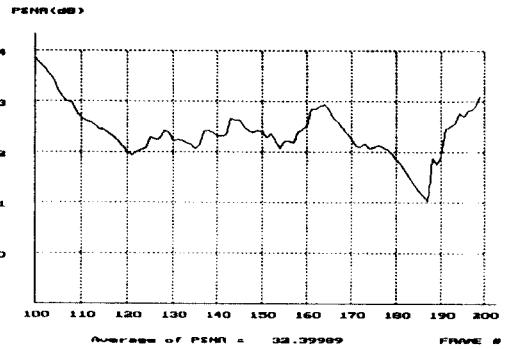


그림 6.  $TH_m=10$ ,  $TH_S=225$ , Claire 영상에 대한 제안된 동영상 부호화에 따른 PSNR 변화.  
Fig 6.  $TH_m=10$ ,  $TH_S=225$ , PSNR variation according to proposed moving image encoding of Clarie images.

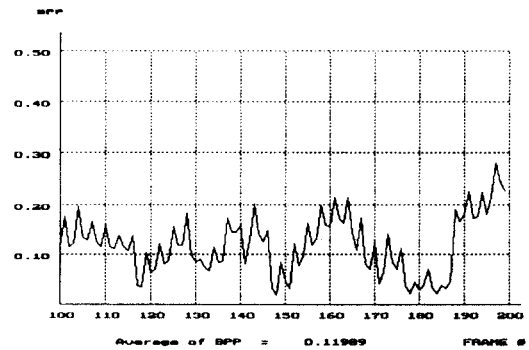


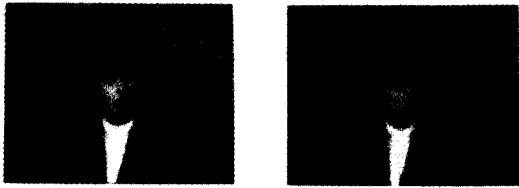
그림 7.  $TH_m=10$ ,  $TH_S=225$ , Claire 영상에 대한 제안된 동영상 부호화에 따른 BPP 변화.  
Fig 7.  $TH_m=10$ ,  $TH_S=225$ , BPP variation according to proposed moving image encoding of Clarie images.



(a) 104번째 프레임  
(a) 104th frame.

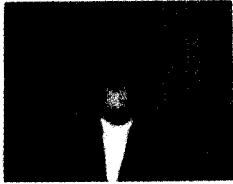


(b) 128번째 프레임  
(b) 128th frame.

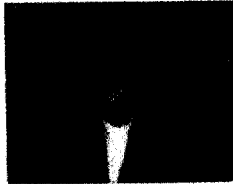


(c) 143번째 프레임  
(c) 143rd frame.

(d) 161번째 프레임  
(d) 161st frame.



(e) 182번째 프레임  
(e) 182nd frame.



(f) 197번째 프레임  
(f) 197th frame.

그림 8. 복원된 Claire 연속 영상 (QCiF,30Hz)  
Fig 8. Decoded Claire successive images. (QCiF,30Hz)

### V. 결 론

본 논문은 연속 영상을 위한 프랙탈 부호화 알고리즘의 가장 근본적인 문제인 부호화 시간에 중점을 두고, QCIF 형태의 144 × 176 크기의 연속 영상에 대해 실시간으로 부호화가 가능한 알고리즘을 제안하였다. 제안된 알고리즘은 50MHz IBM 486 PC에서 Claire, Miss America 그리고 Alex 영상을 이용하여 성능을 테스트하였다. 비트율을 줄이기 위해서 기존의 4 × 4 지역블럭을 8 × 8 블럭으로 확장하였고, 지역블럭의 확장으로 인한 복원 화질의 저하는 쿼드-트리 분할 방식을 적용하여 개선하였다. 8 × 8 지역블럭 분할을 위하여 8 × 8 지역블럭과 공간축소변환된 8 × 8 정의역블럭의 유사성 정도에 따라 블럭 분할하여 블럭 분할로 인한 부호화시간을 최소화시켰다. 또한 지역블럭과 공간축소변환된 정의역블럭의 유사성 정도에 따라 프랙탈 계수의 개수를 변화시켜 화질 저하를 최소화하면서 전송 비트율과 부호화 시간을 단축시켜서 실시간 처리 가능하게 하였다. 각 프레임에 대해

전체 영상을 부호화시 평탄임계치  $TH_5 = 225$ 에서 Claire 영상에 대하여 13.1Hz의 프레임당 속도를 유지하며, 프레임간 에러가 큰 블럭, 즉 움직임이 있는 블럭만을 부호화할 때 30Hz의 빠른 부호화가 가능하였다.

### 참 고 문 헌

1. ISO/IEC JTC1/SC29/WG11, DIS 13818-1, June, 1994.
2. ISO/IEC JTC1/SC29/WG11, CD 13818-2, Nov. 1993.
3. ISO/IEC JTC1/SC29/WG11, CD 13818-3, Nov. 1993.
4. B. Mandelbrot, *The Fractal Geometry of Nature*. W. H. Freeman and Co., San Francisco
5. M. F. Barnsley, *Fractals Everywhere*. Academic Press, 1988.
6. M. F. Barnsley and Alan D. Sloan "A better way to compress images," *Byte 13(1)*, 215-224, 1988.
7. M. F. Barnsley and Lyman P. Hurd, *Fractal Image Compression*. AK Prters, Ltd., p244.
8. A. E. Jacquin, *A fractal theory of iterated Markov operators with applications to digital image coding*, Ph. D. dissertation, Georgia Tech, 1989.
9. A. E. Jacquin, "A novel fractal block-coding technique for digital images," *Proc. IEEE ICASSP*, pp. 2225-2228, 1990.
10. A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformation," *IEEE Trans. on Image Processing*, vol. 1, no. 1, pp. 18-30, Jan, 1992.
11. G. E. Oien, S. Lepsoy and T. A. Ramstad, "An inner product space approach to image coding by contractive transforms," *Proc. ICASSP '91*, pp. 2773-2776, 1991.
12. S. Lepsoy, G. E. Oien and T. A. Ramstad, "Attractor image compression with a fast non-iterative decoding algorithm," *Proc. IEEE ICASSP '93*, vol. 5, pp. 337-340, Apr. 1993.
13. G. E. Oien, *L<sub>2</sub>-optimal attractor image coding with fast decoder convergence*. Ph. D. thesis, The

Norwegian Institute of Technology, May 1993.

14. Y. Fisher, E. W. Jacobs and R. D. Boss, "Fractal image compression using iterated transforms," *Technical Report*, Naval Ocean System Center, San Diego, CA92142-5000.
15. D. M. Monro and F. Dudbridge, "Fractal block coding of images," *Electronics Letters*, Vol. 28, no. 11, pp 1053-1055, 21 May 1992.
16. D. M. Monro, "A hybrid fractal transform," *Proc. IEEE ICASSP '93*, pp. 169-172, 1993.
17. D. M. Monro and F. Dudbridge, "Fractal approximation of image blocks," *Proc. IEEE ICASSP '92*, vol. 3, pp. 485-488, 1992.
18. E. Reusens, "Sequence coding based on the fractal theory of iterated transformation systems," *Proc. SPIE VICEP '93*, vol. 2094, pp. 132-140, 1993.
19. B. Hürtgen and P. B. ttgen, "Fractal approach to low rate video coding," *Proc. SPIE VICEP '93*, vol. 2094, pp. 120-131, 1993.
20. B. H rtgen and C. Stiller, "Fast hierarchical codebook search for fractal coding of still images," *Proc. SPIE VICEP '93*, vol. 1977, pp. 397-408, 1993.
21. D. L. Wilson, J. A. Nicholls, and D. M. Monro, "Rate Buffered fractal video," *Proc. IEEE ICASSP '94*, pp. 505-508, 1994.
22. H. Peitgen, H. Jurgén, D. Saupen, *Chaos and Fractal New Frontiers of Science*, Springer Verlag, New York, 1992.
23. Greg Viens and Monson H. Hayes, III, "Adaptive IFS image coding with proximity maps," *Proc. IEEE ICASSP-93*, pp. 349-352, 1993.
24. R. L. Burden and J. D. Fairs, *Numerical Analysis third edition*, PWS Publishers, p676, 1985.
25. 도원, 서정태, 윤대회, "가변블럭적용 프랙탈영상 압축 알고리즘," 제7회 신호처리합동학술대회 논문집 제7권 1호, 1994.
26. Y. Fisher, *Fractal Image Compression: Theory And Application*, Springer-Verlag, New York, 1994.



金 載 喆(Jae Chul Kim) 정회원  
 1964년생  
 1990년:경북대학교 전자공학과 (학사)  
 1992년:경북대학교 전자공학과 (석사)  
 1993년~현재:경북대학교 전자공학과 박사과정

※주관심분야: 실시간 영상처리, 마이크로프로세서 응용



朴 宗 植(Jong Sik Park) 정회원  
 1953년생  
 1976년:서울대학교 물리학과(학사)  
 1978년:한국과학기술원 수학 및 물리학과(석사)  
 1987년:University of Florida, 전기공학과(박사)

1987년~현재:경북대학교 전자전기공학부 교수  
 ※주관심분야: VLSI 설계, DSP