

프레임간 차신호의 블록 분할을 이용한 프랙탈 동영상 부호화

정회원 박용기*, 박철우**, 박재운**

Fractal Video Coding Using Block Partition of Frame Difference Signal

Yong-Ki Park*, Chul-Woo Park**, Jae-Woon Park** *Regular Members*

요 약

본 논문에서는 H.261, 또는 MPEG에서 동영상의 표준으로 사용되고 있는 DCT기법 대신 프랙탈 이론에 근거하여 동영상의 부호화 시간을 단축하면서 초저속 통신을 할 수 있는 압축 기법을 제시하였다. 먼저 첫번째 프레임과 두번째 프레임간의 차영상 블록을 구하고, 두 가지의 임계값, 즉 임계값 Th1과 임계값 Th2로 움직임 블록을 찾아낸 다음, 그 블록의 RMS값과 분산값으로 프랙탈 계수를 구한다. 제안한 방법으로 실험한 결과 기존의 선형 근사화 방식보다 압축율은 향상되었다. 반면 PSNR값은 1dB~1.2dB 정도 낮아졌으나 주관적인 화질에는 큰 차를 느끼지 못하였다.

ABSTRACT

This paper presents a compression technique based on fractal theory for fast encoding image sequences for very low bit-rates image communication instead of DCT coding which is a standard compression method of moving image used in the H.261, or MPEG. We calculated RMS values by blocks between the first and the second frames. We chose two threshold values, i.e. The first threshold is Th1 and second threshold is Th2. We found blocks with movement and found fractal coefficients by comparing RMS values and two threshold values. Experimental results show that the compression rate is improved compare to liner approximation and PSNR reduced from 1dB to 1.2dB, but the subjective fidelity remained almost unchanged.

I. 서 론

영상 신호를 디지털 데이터로 전송하려면 기저 대역의 대역폭이 4~6MHz정도이므로 Nyquist 샘플링을 했을 때 샘플당 8비트로 표현하면 발생하는 데이터량은 64~96MHz로 대단히 많다. 따라서 이와 같이 디지털화된 영상 신호를 전송하기 위해서는 넓은 전송 대역폭이 필요하다^[1]. 이러한 넓은 전송 대역폭은 통신 측면에서 여러 가지로 불리하고, 특히 위성 통신과 같이 제한된 채널 용량을 갖는

전송선로인 경우에는 넓은 전송폭이 문제가 되므로^[2] 전송 대역폭을 줄이기 위한 데이터 압축에 관한 연구가 많이 되어 왔다. 특히 동영상인 경우 영상을 단순히 디지털화하여 사용하면 다량의 데이터로 인해 대용량의 메모리 공간이 필요하고 디지털 TV방송이나 영상회의에서는 전송 시간이 많아지므로 실시간 처리가 어려워진다. 따라서 동영상 데이터를 효율적으로 부호화할 수 있는 새로운 방법들이 연구되고 있다^[3-12]. 데이터 압축 방식은 수행 방식에 따라 예측 방식과 변환 방식, 이들의 결합에 의한 복합 방식으로 나눌 수 있으며, 구성 형태에 따라

* 양산대학 컴퓨터응용과, 정회원.

** 동부산대학 정보통신과, 정회원

논문번호 : 99004-0315, 접수일자 : 1999년 3월 15일

Interframe 부호화 방식과 Intraframe 부호화 방식으로 나누어진다. 예측 방식은 샘플 사이의 상관 관계를 이용하여 중복성을 줄여 압축하는 방식이고, 변환 방식은 상관 관계가 많은 화소들로부터 특정한 변환을 통해 서로가 독립적인 값을 갖도록 한 다음, 변환 영역에서 기본 함수의 계수들 가운데서 에너지가 적은 계수들은 버리고 에너지가 큰 계수들만을 전송함으로써 압축하는 방식이다.

그리고 Interframe(프레임내)부호화 방식은 공간 영역에서 중복성을 줄이는 방법이며, Intraframe(프레임간)부호화 방식은 시간 영역에서 중복성을 줄여 데이터를 압축하는 방식이다. 최근 들어 반도체 기술의 발달로 하드웨어 구현이 용이해지면서 Interframe 부호화 방식에 많은 연구가 진행되고 있다. H. 261, MPEG과 같은 동영상 압축 시스템 표준은 기존의 제한된 전송률을 이용하여 초저속 전송이 요구되는 응용분야에는 적용되기 어렵다. 이것을 해결할 수 있는 방법으로 프랙탈 부호화 기법^[10]이 고려되고 있으며 저속 채널의 경우 시간과 공간의 종속성을 설명하기 적합한 하이브리드 부호화 기법^{[10][11]}이 채택되고 있다. 동영상 부호화는 시간적 중복성을 제거하는 부호화 하는 기법으로 시간적으로 연속하는 영상들 간에는 움직임이 있는 부분을 제외하고는 많은 중복성을 가진다. 따라서 이러한 중복성을 제거하기 위해 가장 많이 연구되는 것이 움직임 보상 예측 부호화 기법^{[10][12]}이다. 이 기법에서 움직임을 추정하는 방법으로는 화소 단위 알고리즘(PRA)과 블록 정합 알고리즘(BMA)^[12]이 있다. 움직임 추정 방법은 영상의 변화에 따라 정확한 움직임을 얻을 수 없으며 영상내의 모든 시간적 중복성을 제거하기는 어렵다. 따라서 움직임 추정시 발생한 예측 오차에 의해 화질이 저하되고 경계 효과가 발생하므로 이를 보정하기 위해서는 예측 오차 부호화가 필요하다. CCITT와 MPEG에서는 예측 오차 부호화 방법으로 DCT를 사용하도록 하고 있으나 이 방법은 신호의 공간적 중복성을 제거하는 정지 영상 부호화 기법으로 공간적 상관 관계가 매우 낮은 예측 오차 신호에서는 압축 효과가 적다. 이 문제를 해결하기 위해 DCT기법 이외에 서브밴드 부호화, 피라미드 부호화 등과 같은 기존의 영상 부호화 방법을 이용하여 예측오차에 대한 부호화 효율을 높이고자 하였으나 큰 효과를 거두지 못하였다. 따라서 본 논문에서는 예측오차 자체를 부호화하지 않고 움직임 보상 예측 부호화에 기반을 둔 프랙탈 부호화 방법을 제안하였다.

본 연구에서는 먼저 이전 프레임의 복원 영상과 새로 입력되는 프레임간의 차영상을 구하여 블록 분할한다. 분할된 각 블록에 대해 RMS를 계산하여 허용된 임계치를 초과하는 블록은 움직임이 있는 블록, 즉 움직임 블록으로 분류한다.

이러한 움직임 블록은 프랙탈 부호화되어야 하며 탐색 대상인 Domain영역은 이전 프레임의 복원 영상이 된다. 이전 프레임에서 이용되는 Domain블록의 위치 정보는 움직임 예측에 대한 정보에 해당한다. 이 때 움직임 블록으로 분류된 블록에 대해서만 분산을 구하여 분산값이 기준치 이상이 될 때 재분할하여 블록을 가변시켜 부호화함으로써 복원 영상의 화질을 향상시켰고, RMS 임계치와 분산 기준치를 초과하지 않는 블록은 움직임이 없는 블록으로 분류하여 부호화 대상에서 제외시켜 압축율을 높일 수 있었다.

II. 이론적 배경

2.1. 동영상의 수학적 이론

동영상 부호화에서의 완전 척도 공간을 (F, d)라 하면, 척도 d는 다음과 같다.

$$\begin{aligned} \forall \mu, \nu \in F, d(\mu, \nu) \\ = \frac{1}{KLM} \sum_{i=1}^K \sum_{j=0}^L \sum_{k=0}^M (\mu_{i,j,k} - \nu_{i,j,k}) \end{aligned} \quad (1)$$

여기서 $\{R_i, 0 \leq i \leq N\}$ 을 시퀀스가 서로 겹치지 않는 분할 블록인 N개의 Range블록들로 정의하면 변환 F는 식(2)와 같이 나타낼 수 있다.

$$\begin{aligned} w(\mu) = \sum_{0 \leq i \leq N} w_i(\mu |_{R_i}) = \sum_{0 \leq i \leq N} w_i(\mu |_{D_i}) \\ \text{with } w_i : D_i \rightarrow R_i, \forall \mu \in F \end{aligned} \quad (2)$$

여기서 $\mu |_{R_i}$ 는 시퀀스 μ 를 Range 블록 R_i 로 한정하는 것을 의미한다. R_i 와 유사한 블록을 탐색 방법으로는 프레임내 탐색과 프레임간 탐색이 있다. 프레임내 탐색은 공간적인 자기유사성을 이용하고 프레임간 탐색으로는 시간적인 중복성을 제거한다. 이 때 두 가지 탐색 방법은 처리 과정이 다르므로 명백히 구분되어야 한다.

동영상에서 원영상의 시퀀스 μ_{orig} 와 최종 끝개 μ_0 간의 공간적인 자기유사성을 측정할 수 있는 척도는 식(3)과 같다.

$$\lim_{n \rightarrow \infty} d(\mu_{orig}, w^n(\mu_0)) \leq \frac{1}{1-s} d(\mu_{orig}, w(\mu_{orig})) \quad (3)$$

식 (3)을 기하변환 G_i 와 수치변환 M_i 로 표시하면 다음과 같다.

$$w_i = M_i \circ G_i$$

이 때 기하변환 G_i 는 Domain블록을 Range블록의 크기로 축소시키는 R_i 와 R_i 에 의해서 축소된 Domain블록을 Range블록으로 매핑하는 Suffle변환 S_i 로 구분된다. 따라서 위의 식은 다음과 같이 표시할 수 있다.

$$w_i = M_i \circ S_i \circ R_i$$

기하 변환은 Domain블록이 Range블록에 매핑되는 방법을 결정하고 Suffle변환은 미리 정해진 8가지 변환을 사용한다. 수치변환은 기하 변환이 이루어진 Domain블록이 Range블록과 유사성을 가지도록 화소값을 변경시킨다. 여기서 고려하는 매개변수는 8가지 Suffle변환과 Contrast scale a , 밝기의 변화 Δg 이다. 변환 w_i 의 수치 변환 M_i 는 다음과 같이 표시할 수 있다.

$$M_i \begin{pmatrix} x \\ y \\ t \\ z \end{pmatrix} = \begin{pmatrix} a_i & b_i & 0 & 0 \\ c_i & d_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & a \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \Delta g \end{pmatrix} \quad (4)$$

이 식은 두 영상간의 유사성을 표현하며 a_i, b_i, c_i, d_i 는 기하 변환과 Suffle변환을 나타내고 $|a|$ 는 Contrast scaling으로 축소 조건에 의해 1보다 작아야 하며 Δg 는 Range영역과 Domain영역의 평균 화소값의 차인 Offset을 나타낸다. 영상 부호화는 식(4)의 변환 M_i 의 각 계수를 구하는 과정으로 볼 수 있다. 최상의 변환 매개변수를 구하기 위해 R_i 와 같은 프레임내에 있는 가능한 모든 Domain블록 D_i 를 탐색하며 이 때 기하 변환은 Domain블록을 Range블록으로 매핑한다. 축소 변환 이론을 적용하여 동영상의 시간적 중복성을 제거하려면 축소 조건 때문에 쉽지가 않다. 시간적 중복성은 프레임간의 유사성에서 발생하고 같은 Scale을 가지므로 변환 w_i 는 1에 가까운 축소율을 가진다. 이 때 식 (4)에서 $\frac{1}{1-s} \rightarrow \infty$ 가 되므로 끝개와 원영상 시퀀스간의 차이는 무한대로 된다. 이러한 문제를 극복하기 위해 변환 w_i 에서 기하변환을 삭제하여 시간

방향의 탐색 방법을 다르게 한다. 시간적인 중복성을 제거하기 위해 사용되는 식은 다음과 같다.

$$w_i \begin{pmatrix} x \\ y \\ t \\ z \end{pmatrix} = \begin{pmatrix} x \\ y \\ t \\ z \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta t \\ 0 \end{pmatrix} \quad (5)$$

여기서 Δt 는 음수로 -1로 고정하고 매개 변수 Δx 와 Δy 는 Range블록 R_i 와 최적으로 정합하는 Domain블록 D_i 의 탐색에 의해 결정된다. 부호화에서 이러한 시간적 중복성을 이용하면 3차원 블록 정합의 비교 수를 줄일 수 있다. 식 (5)의 변환 w_i 는 축소 조건이 만족되지 않은 상태에서 D_i 의 시간 방향 탐색으로 변환 매개 변수를 결정하더라도 복원과정에서 고정점에 수렴하게 된다. Δt 는 음수이므로 마지막 프레임은 IFS의 자기유사성에 의해 부호화된다. 이러한 시간 중복성에 의해 부호화된 모든 영역은 다른 프레임에서 직접, 간접적으로 정의될 수 있으며 최종 프레임은 별도로 프레임내, 또는 참조 프레임으로 표현할 수 있다.

2.2. 동영상의 적응적 블록 분할

동영상 부호화에서 적응적 블록 분할법을 적용하면 비트율 조정이 가능해지므로 메모리 감소나 화질을 임계 매개 변수로 조정할 수 있다. 또한 분할된 블록은 영상의 종류에만 영향을 받으므로 시간이나 공간 양쪽 다 블록의 통계적 성질에 따라서 블록의 크기를 최상의 방법으로 가변할 수 있다. 분할은 트리 구조로 반복적으로 행해지며 트리의 루트는 전체 시퀀스가 되고 마지막 노드가 분할 블록으로 부호화된다. 블록 분할점은 $w_i(D_i)$ 에 의해 근사화된 최대 평균 자승 오차의 허용오차로 결정한다. 이러한 분할은 시간축이나 공간축에 따라 행해질 수 있다. 시간 분할에서는 두 개의 서브블록을 만들고 그 크기는 이전의 계층적 레벨 블록 크기의 1/2이 된다. 이 때 공간의 크기는 같은 크기를 유지한다. 공간 분할에서는 공간의 크기를 4개의 서브블록으로 분할하지만 시간축의 크기는 앞 레벨의 크기와 매칭된다. 이러한 과정이 그림 1에 나타나 있다. 블록의 선택은 공간 분할에 의한 4개의 서브블록의 근사화에 의한 전역 평균자승오차가 시간 분할에 의해 생성된 두 개의 서브블록의 근사화에 의한 평균자승오차와 비교되고 가장 매칭이 잘 되는 블록이 선택된다. 이러한 과정을 식으로 나타내면 식 (6)와 같다.

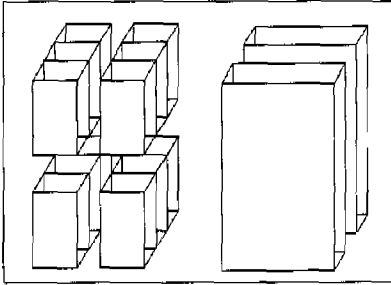


그림 1. 공간 분할과 시간 분할

$$K1 = \sum_{i=1}^2 d(w_i, D_i^s, R_i^s) \quad (6)$$

$$K2 = \sum_{i=1}^2 d(w_i, D_i^t, R_i^t)$$

$$split = \begin{cases} Spatial & \text{if } K1 < K2 \\ Temporal & \text{otherwise} \end{cases}$$

여기서 s 는 공간 분할에 의해 생긴 블록을, t 는 시간 분할에 의해 생긴 블록을 의미한다.

III. 동영상 부호화 방법

3.1. 블록 정합에 의한 움직임 예측

이동보상 예측 부호화에서 블록 정합 알고리즘은 시간적으로 이웃하는 두 프레임을 각각 일정한 크기의 블록으로 분할하여 각 블록마다 평행 이동 움직임을 모델로 하여 움직임을 찾는 방법으로 블록 내의 모든 화소는 동일한 움직임을 갖는다. 이 때 블록이 동일한 움직임을 갖는 물체(Object)의 한 부분이면 정확한 움직임을 찾지만 블록이 움직임이 있는 두 물체의 경계 부분에 있으면 정확한 움직임을 찾지 못한다. 따라서 움직임이 있을 때 블록간의 상호 상관성을 측정하는 기준이 필요하며 식 (7)와 같은 평균제곱오차(MSE)나 평균절대오차(MAD)를 이용한다.

$$MSE(n, m) = \frac{1}{LM} \sum_{x=0}^{L-1} \sum_{y=0}^{M-1} [f_x(x, y) - f_{j-1}(x-n, y-m)]^2 \quad (7)$$

$$MAD(n, m) = \frac{1}{LM} \sum_{x=0}^{L-1} \sum_{y=0}^{M-1} |f_x(x, y) - f_{j-1}(x-n, y-m)|$$

여기서 f_k 와 f_{k-1} 은 k 와 $k-1$ 번째 프레임의 $L \times M$ 영상 블록을 나타내고 n, m 은 블록의 움직임을 나타내는 변위를 말한다. 블록의 움직임 탐색 범위는 블록의 크기가 $M \times M$ 일 때 $(M+2p) \times$

$(M+2p)$ 크기의 탐색 영역을 설정하고 이 영역 내에서 블록 이동을 탐색한다. 이 때 탐색 범위를 전역 탐색으로 하면 부호화 시간은 많이 걸리지만 상대적으로 알고리즘이 간단하고 하드웨어 구현이 쉬워진다는 장점이 있다. 그림 2는 움직임보상 예측 부호화에서의 탐색 블록과 탐색 범위를 나타낸 것이다.

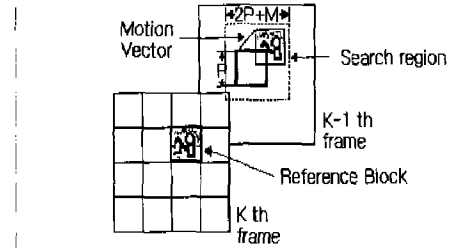


그림 2. 움직임 탐색 설명도

3.2. 예측 오차 부호화

영상내의 움직임이 있는 블록은 움직임 추정기법으로 그 위치를 알아 낼 수 있으나 블록의 회전이나 소멸 등으로 인해 원영상과 움직임 추정 방법을 통한 복원영상 사이에 예측 오차가 발생한다. 기존의 부호화 방법에서는 움직임 추정 방법을 이용하여 움직임을 찾고 예측 오차를 DCT로 부호화한다. DCT를 이용한 영상 부호화는 영상을 2차원으로 DCT한 후 변환계수를 양자화하여 정보를 전송한다. 2차원 DCT와 IDCT는 식 (8)과 같다.

DCT는 8×8 블록에 적용되고 변환 계수는 run-length 부호화로 양자화한다. 양자화는 움직임 보상후의 차신호의 에너지가 주파수 대역에 일정하게 분포하고 있으므로 다음과 같이 균등 양자화한다.

$$\begin{aligned} c(i, j) &= (16 \cdot C(i, j)) // W(i, j) \\ QC(i, j) &= C(i, j) // (2 \cdot QP) \\ F(u, v) &= 2c(u)c(v) \sum_{j=0}^{M-1} \sum_{k=0}^{M-1} f(j, k) \\ \cos \left[\frac{(2i+1)u\pi}{2M} \right] \cos \left[\frac{(2k+1)v\pi}{2M} \right] \\ f(u, v) &= \frac{2}{M} \sum_{j=0}^{M-1} \sum_{k=0}^{M-1} c(u)c(v)F(u, v) \\ \cos \left[\frac{(2i+1)u\pi}{2M} \right] \cos \left[\frac{(2k+1)v\pi}{2M} \right] \\ c(w) &= \begin{cases} \frac{1}{\sqrt{2}} & \text{for } w=0 \\ 1 & \text{for } w=1, 2, \dots, M-1 \end{cases} \\ u, v &= 0, 1, \dots, M-1 \end{aligned} \quad (8)$$

이 때 $C(i,j)$: 양자화전의 변환 계수, $QC(i,j)$: 양자화 후의 계수, $W(i,j)$: 화소의 가중치 행렬, // 는 나눗셈한 후 소수 첫자리에서 반올림하는 연산자, QP 는 양자화 scale로 1에서 31사이의 값으로서 양자화 효과와 할당되는 비트수를 변화시킨다. 양자화된 계수값들은 -255에서 255까지의 값으로 Clipping한다. 그림 3은 DCT를 이용한 동영상 부호화의 과정을 나타낸다.

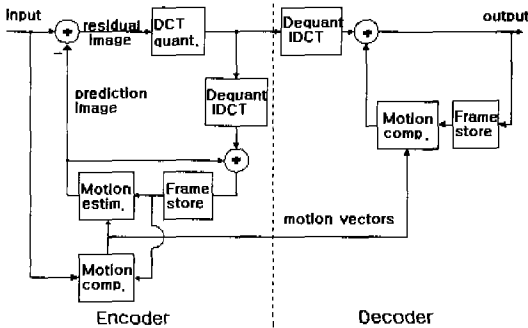


그림 3. DCT를 이용한 동영상 부호화 과정

IV. 제안된 프랙탈 동영상 부호화

4.1. 압축률 및 부호화 시간 개선

제안된 가변 블록 분할을 이용한 프랙탈 부호화에 대한 블록도는 그림 4와 같다. 연속된 프레임의 경우는, 그림 5에서 보는바와 같이 프랙탈 부호화해야 할 움직임 블록을 찾기 위해 이전 프레임과 새로 입력된 프레임간의 차영상을 구하고, 그 차영상을 블록단위로 분할한다.

첫 번째 프레임과 두 번째 프레임에서 블록단위로 RMS값의 차를 구한 다음 두 가지의 임계값을 주어서 움직임 블록을 찾아 Parent 블록으로 만든다. 첫 번째 임계값을 $Th1$, 두 번째 임계값을 $Th2$ 라면 두 영상간의 블록 RMS값이 $Th1$ 보다 적은 경우에는 움직임이 없는 블록으로 판단하여 프랙탈 계수를 구하지 않고, RMS값이 $Th1, Th2$ 사이에 있을 경우에는 움직임이 있는 것으로 판단하여 Parent 블록 단위로 프랙탈 부호화를 한다.

RMS값이 $Th2$ 보다 클 경우에는 움직임이 심한 경우로 판단하여 Parent 블록은 4개의 Tile-블록으로 분할된다. 그리고 분할된 각각의 Tile 블록(새로 입력된 프레임 영상)이 Domain 영역(이전 프레임의 복원 영상)에서 유사영역을 찾아 프랙탈 계수를 구한다. 이 과정에서 새로 입력되는 프레임 전체 영역

에서 부호화되는 블록은 단지 움직임 있는 블록들만 부호화되므로 압축율을 높이는 효과를 얻을 수 있다. 부호화된 움직임 블록들의 프랙탈 계수의 정보는 디코딩 과정으로 전송한다. 디코딩부에서는 이전 프레임의 복원 영상을 Domain 영상으로 정한 뒤, 전송된 프랙탈 계수와 움직임 블록의 위치를 이용하여 Domain-블록에서 움직임 블록에 해당하는 위치의 블록을 Temp 블록으로 잡는다. 그리고 그 Temp 블록의 x, y 좌표와 밝기값에 프랙탈 계수를 반복하여 곱해줌으로 그 위치에 해당하는 두 번째 프레임의 Parent 블록을 구하게 된다. 이 때 첫 프레임은 정지 영상 기법으로 부호화하며 전 영역의 블록에 대해 부호화하여 프랙탈 계수를 구해야 한다. 반면에 두 번째 프레임부터는 이전 프레임과 새로 입력된 프레임간의 움직임 블록만 찾아 부호화함으로써 프레임간의 시간적 중복성을 제거한다. 부호화한 프랙탈 정보를 양자화할 때 $Th1 < RMS < Th2$ 인 경우에는 Parent 블록 크기로, $Th2 < RMS$ 인 경우에는 부블록 크기로 양자화한다. 양자화할 때의 할당된 비트에 대한 설명은 표 1과 같다.

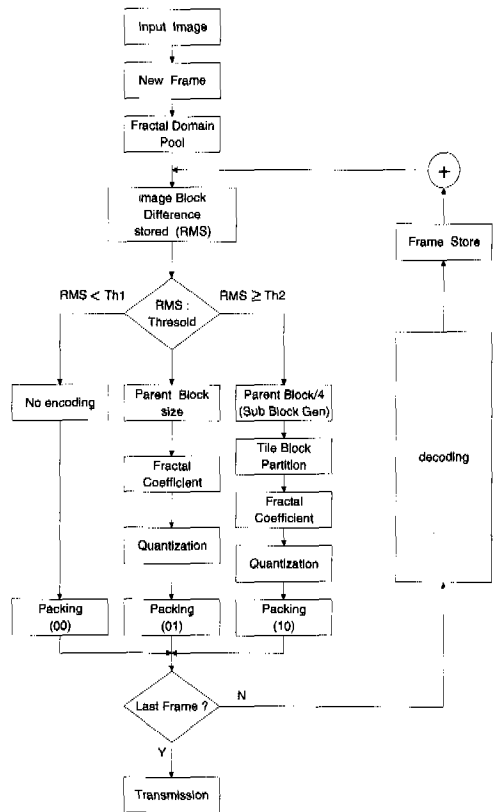


그림 4. 가변블록 분할을 이용한 개선된 방법의 블록도

표 1. 할당된 비트에 대한 설명

| | |
|---|--|
| a | 하나의 프레임에서 프랙탈 부호화한 정보를 계속 해서 양자화할 것을 나타내는 비트 |
| b | 해당블록의 위치를 암시하는 Block Number |
| d | 16블록, 8블록을 구분하는 비트 |
| e | Tile Block의 x좌표측 프랙탈정보를 담은비트 |
| f | Tile Block의 y좌표측 프랙탈정보를 담은비트 |
| g | Tile Block의 scale 프랙탈정보를 담은비트 |
| h | Tile Block의 offset 프랙탈정보를 담은비트 |

이제 부호화에서 생성된 움직임 블록의 x, y좌표 값과 프랙탈 계수를 양자화하는 과정을 거쳐야한다. 그림 5와 그림 6은 각각 Parent 블록 크기로 분할된 경우와 부분블록으로 분할된 경우의 양자화 테이블이다. 이와같이 양자화된 값들은 디코딩부에서 해석되어 움직임 블록의 좌표값에 의해 구해진 위치에 덮어 씌움으로써 원영상을 복원해 낸다.

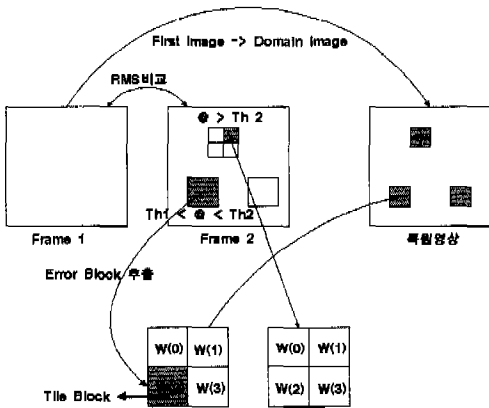


그림 5. 연속된 프레임간의 Error Block만의 프랙탈 압축을 통한 영상복원

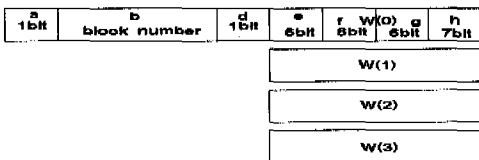


그림 6. Parent Block의 양자화 테이블

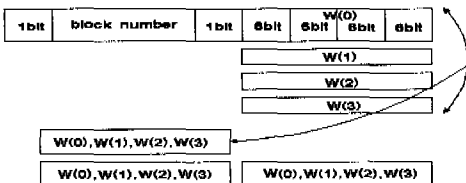


그림 7. 부분블록들의 양자화 테이블

4.2. 동영상 프랙탈 계수의 양자화

제안된 방법에서는 이전 프레임과 새로 입력되는 프레임에서 움직임이 있는 블록만을 탐색하여 프랙탈 계수값을 구하고, 그 값을 비트 단위로 양자화한다. 부호화 과정에서 전달되는 매개 변수의 블록당 비트 할당 구조는 다음과 같이 3가지 방법을 택하였다. 첫 번째 방법은 Pack의 실행 유무를 결정하는 1비트, 움직임 블록의 블록 순서를 표시하는 12비트, 블록 분할 여부에 1비트, 마지막 파일 끝을 의미하는 1비트, 프랙탈 변환계수의 비트 할당 방법, 두 번째 방법은 프레임의 전 블록에 대해 움직임이 있는 블록인지 아닌지를 결정하고 Parent 블록 혹은 부분블록으로 분할됨을 결정하는 2비트와 그에 따른 프랙탈 변환계수 비트 할당 방법, 세 번째 방법은 움직임 블록인지 아닌지를 결정하고 Parent 블록 크기 혹은 부분블록으로 분할됨을 결정하는 2비트와 움직임이 없는 경우에 대한 갯수를 표시하는 8비트와 마지막 파일 끝을 표시하는 2비트, 프랙탈 변환 계수의 비트 할당으로 구성하는 방법이다. 위 3가지 방법에서 프랙탈 변환 계수의 비트는 6, 6, 6, 7 비트로 할당하였다. 다음은 352 * 288 크기의 입력 영상에 대하여 parent 블록의 크기를 16으로 하여 Pack할 때 구해지는 Error 블록의 번호에 의해 x, y 좌표값을 구하는 방법이다.

$$x = \text{block_number} \% (\text{영상의}x \text{ size} \div \text{parent block}) \times \text{parent block size}$$

$$y = \text{block_number} / (\text{영상의}x \text{ size} \div \text{parent block}) \times \text{parent block size}$$

여러 블록의 번호가 3번인 경우 위의 식에 의해 좌표는 (48, 0)이 된다. 그림 8은 Pack하는 방법을 나타낸다.

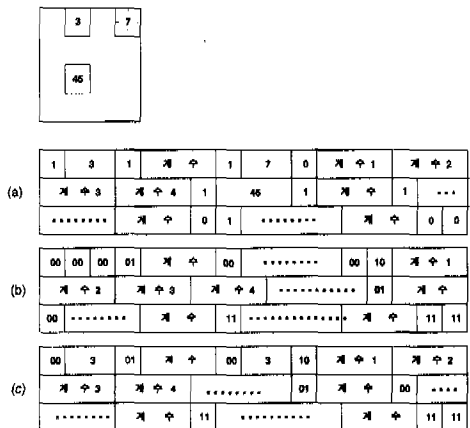


그림 8. 프랙탈 계수를 Pack하는 방법

그림 8의 (b)경우는 움직임이 없는 블록 수만큼 00으로 채워야하므로 (a), (c)경우와 같이 블록에 비트를 할당하면 압축율을 높일 수 있다. 특히 움직임이 적은 영상에서 얻는 압축율은 더욱 차이가 심할 것이다.

V. 실험 결과 및 검토

본 논문의 실험에서는 고정 크기 블록 분할 기법에 의한 동영상 부호화 성능과 차신호에 의한 가변 블록 분할 기법에 의한 성능을 비교 검토하였다. 실험 환경은 Dec-Workstation으로 부호화 시간은 모든 작업 환경을 포함한 상대적인 시간이므로 사용하는 실험기기의 종류에 따라 달라질 수 있다. 실험 결과에 대한 성능 비교의 기준으로는 압축비, PSNR, bpp, 총 비트량 등을 사용하였다. 실험 영상은 Miss America 영상 시퀀스 352×288크기이다. 최소 허용 임계값은 8, 최대 허용 임계값은 10으로 정하고, 디코딩의 반복 횟수는 10번으로 하여 실험하였다. 표2는 Range 블록을 8x8, 4x4로 하고 처음의 16개의 프레임에 대상으로 고정 크기 선형 변환 방법과 제안한 방법으로 실험한 결과이다. 제안한 방법은 차영상을 이용하여 움직임 블록을 RMS값에 의해서만 판정하도록 하는 방법이며, 제안한 방법2는 RMS값에 의해 탐색된 움직임 블록의 분할 여부를 판정하기 위해 블록의 분산을 이용하는 방법이다. 방법2의 실험에 적용한 RMS값은 10, 분산값은 150이다.

표 2. 가변블록을 사용한 경우의 성능 비교 (Range block 8*8, 4*4)

| | linear | proposed method | |
|----------------------|--------|-----------------|----------|
| | | method 1 | method 2 |
| compression rate (%) | 15.75 | 76.3015 | 92.7343 |
| PSNR | 35.53 | 34.4838 | 34.314 |
| Total-Byte | 261035 | 19927 | 17491 |



(a) 7 frame (b) 9 frame

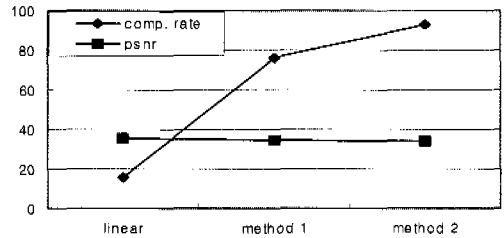
그림 9. 제안한 방법 1에 의한 복원영상

그림 9와 그림 10은 제안한 방법1과 방법2에 의한 복원된 영상 중 7, 9 프레임의 결과를 비교한 것이며, 그림 11은 표 2에서의 3가지 방법의 결과들 중 압축비와 PSNR값을 그래프로 표시한 것이다.

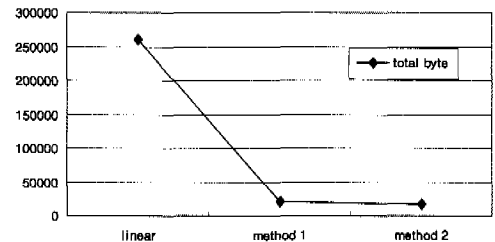


(a) 7 frame (b) 9 frame

그림 10. 제안한 방법 2에 의한 복원영상



(a) compression rate and PSNR value



(b) Total byte value

그림 11. 세가지 방법에 의한 결과 비교

VI. 결론

본 논문에서는 프레임간의 차영상을 이용하여 부호화 시간을 단축하고 화질을 좋게 하면서 압축율을 향상시키는 방법을 제시하였다. 먼저 이전 프레임과 새로 입력되는 프레임간의 차영상을 구하고 그 차영상을 분할한 다음, 각 블록 단위로 구한 RMS값을 두 가지의 임계값 Th1, Th2와 비교하여

움직임 블록을 찾아내고, 그 블록의 분산으로 블록 재분할을 결정한다. 즉 두 프레임간의 블록 RMS 값이 Th1보다 적은 경우에는 움직임이 없는 것으로 판단하여 새로운 프랙탈 부호화 계수를 구하지 않고 이전에 복원된 블록 영상을 그대로 사용한다. RMS값이 Th1, Th2사이에 있을 경우에는 움직임이 있는 것으로 판단하여 분할없이 Parent 블록단위로 프랙탈 부호화를 하고, RMS값이 Th2보다 클 경우에는 움직임이 심한 경우로 판단하여 Parent 블록을 4개의 부분블록으로 분할한 후 새로 입력된 프레임에 있는 각 부분블록의 Tile 블록과 이전 프레임에서 복원된 영상인 Domain 영역에서 프랙탈 계수를 구한다. 결국 연속하여 입력되는 각 프레임 전체 영역을 프랙탈 부호화하는 것이 아니라 첫 번째 프레임만 정지 영상 부호화 기법으로 부호화, 복호화하고, 두 번째 프레임부터는 움직임 블록만을 구하여 부호화하고 복호화함으로써 프레임간의 시간적 중복성을 제거하였다. 또한 압축율을 높이는 방법으로 움직임 블록을 비트 단위로 양자화할 때 세 가지 형태의 비트 할당 방식을 적용하고 성능을 비교하였다. 그리고 같은 압축율을 유지하면서 주관적 화질을 개선하는 방법으로 분산을 이용하여 움직임이 심한 블록일지라도 블록 전체가 일정하게 움직인 경우는 블록 분할을 하지 않도록 하여 화질을 비슷하게 유지시키고 처리 시간도 줄였다. 제안한 방법으로 실험한 결과 기존의 선형 근사화 방식보다 압축율은 4배 이상 향상되었다. 반면 PSNR값은 1dB~1.2 dB 정도 낮아졌으나 주관적인 화질에는 큰 차를 느끼지 못하였다. 따라서 제안한 방법으로 움직임이 있는 블록만을 프랙탈 부호화하면 전체 시스템의 계산량을 줄이고 저속으로 정보를 전송할 수 있으므로 움직임이 크지 않는 영상전화, 영상회의등에 이용할 수 있으리라 기대되며 앞으로는 컬러 동영상을 처리할 수 있는 방법이 연구되어야 할 것으로 본다.

참 고 문 헌

[1] A.K.Jain, "Image Data Compression :A Review", Proceedings of the IEEE Vol.69, No.3, March 1981.
 [2] James Martin, "Communications Satellite systems." Ch13, Prentice - Hall Inc, 1978.
 [3] M.F.Barnsley, *Fractals Everywhere*, Academic Press, 1988.

[4] M. F. Barnsley, L. P. Hurd, *Fractal Image Compression*, AK Peters: Wellesley, Massachusetts, 1993.
 [5] Monro, D. M., "A hybrid fractal transform", Proc. ICASSP 1993, pp. V:169-172.
 [6] Woolley, S. J. Monro, D. M., "Optimum parameters for hybrid fractal image coding", Proc.ICASSP 1995, pp.2571-2574.
 [7] Jacquin,A.E., "A novel fractal block-coding technique for digital images", Proc.ICASSP 1990, pp.2225-2228.
 [8] A.Jacquin, "Image Coding Based on a fractal Theoryof Iterated Contractive Image Transformation," IEEE Trans. Image Processing,vol.1, no.1, pp.18-30, January 1992.
 [9] Bernd Hürtgenand Peter Büttgen, "Fractal approach to low rate video coding", Visual Communication and Image Proc. SPIE/Vis Com '93, Nov. '93, Boston, MA, USA.
 [10] Motion Picture Experts Group, "MPEG Video Committee Draft", Doc.ISO-IEC / JTC1 / S C29 / WG11 MPEG 92 / NO 245, 1992.
 [11] R.J.Clarke, "On transform coding of motion-compensated difference images", IEEE Proc.I, vol.139, no.3, June1992.
 [12] D.M.Monro,F.Dudbridge, "Fractal approximation of image blocks", Proceedings ICASSP-92, vol.3, pp.485-488, March 1992.

박 용 기(Yong-Ki Park)

정회원



1959년 5월 12일생

1983년: 동아대학교 전자공학과 졸업(공학사)

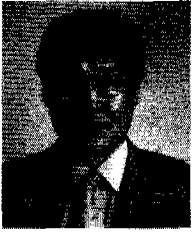
1987년: 동아대학교 대학원 전자공학과 졸업 (공학석사)

1998년: 동아대학교 대학원 전자공학과 졸업(공학박사)

현재: 양산대학 컴퓨터응용과 조교수
 <주관심 분야> 영상처리

박 철 우(Chul-Woo Park)

정회원



1982년 : 동아대학교 전자공학과
(공학사)

1986년 : 동아대학교 대학원
전자공학과(공학석사)

1997년 : 동아대학교 대학원
전자공학과(공학박사)

1989~현재 : 동부산대학 정보통신과 부교수

<주관심 분야> Image Processing, 인공지능

박 재 운(Jae-Woon Park)

정회원



1981년 : 동아대학교 전자공학과
(공학사)

1983년 : 동아대학교 대학원
전자공학과(공학석사)

1994년 : 동아대학교 대학원
전자공학과(공학박사)

1985~현재 : 동부산대학 정보통신과 교수

<주관심 분야> SS통신, GPS