

공간예측에 의한 고속 2×2 프랙탈 영상압축

(A Very Fast 2×2 Fractal Coding By Spatial Prediction)

위 영철 †

(Young Cheul Wee)

요약 본 논문은 극소 원자블록(ultra small atomic block)에 공간예측(spatial prediction)을 적용하여 프랙탈(fractal) 영상압축의 압축시간을 획기적으로 향상시키고 화질/압축률을 향상시키는 방법을 제안한다. 본 방법은 치역(range block)의 크기가 아주 작으면 아주 적은 탐색범위 내에서 변환계수(transformation parameter)들의 값을 극히 제한하더라도 유사한 정의역(domain block)을 쉽게 찾을 수 있고 변환계수들이 좋은 상호관계를 유지함을 이용하여 변환계수 예측으로 화질/압축률을 향상시킨다. 특히, 본 방법은 탐색범위를 극히 제한하기 때문에 기존의 프랙탈 압축방법들 보다 압축시간을 획기적으로 향상시킨다.

키워드 : 프랙탈, 영상압축, 공간예측, 원자블록

Abstract In this paper, we introduce a very fast and efficient fractal coding scheme by using the spatial prediction on ultra-small atomic range blocks. This new approach drastically speeds up the encoding while improving the fidelity and the compression ratio. The affine transformation coefficients between adjacent range blocks induced by this method often have good correlations thereby the compression ratios can further be improved. The proposed method leads to improved rate-distortion performance compared to previously reported pure fractals, and it is faster than other state-of-the-art fractal coding methods.

Key words : fractal, image compression, spatial prediction, atomic block

1. 서론

프랙탈 영상압축은 영상을 서로 겹치지 않는 사각형들로 분할한 각 치역(range block) $R_i = (r_1, r_2, \dots, r_d)$, $1 \leq i \leq n$ 과 유사한 정의역(domain block) $D_{\mathcal{R}i}$ 을 유사 변환(affine transformation)으로 밝기(grey value)를 수정하여 치역 R_i 의 근사치를 구한다 (그림 1 참조). 정의역 $D_{\mathcal{R}i}$ 는 식 (1)에 의하여 $\hat{D}_{\mathcal{R}i} = (\hat{d}_1, \hat{d}_2, \dots, \hat{d}_d)$ 로 변환된다. 복원(decode)의 수렴(convergency)을 보장하기 위하여 정의역의 크기는 가로 세로 각각 치역의 2배로 하고 대비 값은 $|\hat{s}| < 1$ 을 만족하도록 한다. 한 치역과 변환된 정의역과의 오차 $d(R_i, \hat{D}_{\mathcal{R}i})$ 는 식 (2)의 평균제곱오차(mean square error)로 측정한다. 프랙탈 코드(code)는 각 치역의 분할에 사용되는 정보, 정의역의 위치, 유사변환의 계수들로 구성된다.

$$\hat{D}_{\mathcal{R}i} = sD_{\mathcal{R}i} + g \quad (1)$$

$$d(R_i, \hat{D}_{\mathcal{R}i}) = (1/d) \sum_{j=1}^d (r_j - \hat{d}_j)^2 \quad (2)$$

각 치역 R_i , $1 \leq i \leq n$ 에 대한 식 (1)의 변환함수를 ϕ_i 라 하면 프랙탈 코드는 $\mathcal{O} = \phi_1, \phi_2, \dots, \phi_n$ 로 나타낼 수 있다. 임의의 영상 I_0 에서 시작하여 \mathcal{O} 를 반복적으로 적용하면 $\mathcal{O}(I_w) = I_w$ 가 되고 I_w 가 복원된 영상이 된다.

화질은 유사블록과의 오차에 비례하고 유사블록 오차를 줄이기 위해서는 일반적으로 1) 많은 블록을 검색하거나 2) 유사변환의 종류나 계수의 범위를 늘이거나 3) 치역의 크기를 작게 하게 되는데 이에 따라 압축시간이 길어지고 압축률이 저하되게 된다. 전역탐색(full search)과 같은 화질을 유지하면서 압축시간을 향상시키는 알고리즘들은[8-10] 압축시간을 최대 수십 배 줄이는데 그치고 있고 화질을 저하시키면서 압축시간을 줄이는 알고리즘[1-6]들은 고화질에서 압축시간이 현격히 늘어나고 압축률이 저하되는 문제점들을 가지고 있다.

$b \times b$ 유사블록 탐색을 b^2 차원 근접 점 탐색으로 변환하여 압축속도를 개선한 알고리즘[1,2]는 512×512 영상에서 평균 15초 이상 소요된다. 블록합병에 의하여 화

† 종신회원 : 아주대학교 정보 및 컴퓨터공학부 교수
ycwee@madang.ajou.ac.kr
논문접수 : 2004년 7월 19일
심사완료 : 2004년 10월 19일

질/압축률을 개선한 알고리즘[3-5]들은 화질/압축률은 알고리즘[1,2] 보다 우수하지만 더 많은 압축시간을 요구하고 있다. 불필요한 탐색 점을 조기에 제거하여 압축 속도를 개선한 알고리즘[6] 또한 압축시간이 평균 10초 이상 소요되고 있다.

분산(variance) 값의 특성을 이용한 알고리즘[8]은 전역탐색과 같은 화질을 유지하지만 압축시간을 약 4배만 단축하고 이를 개선한 알고리즘[9]는 알고리즘[8]의 속도를 약 2배 개선하는데 그치고 있다. 알고리즘[10]은 위의 방법들에서 분산 값이 유사한 치역에 회전(rotation) 및 반향(reflection)을 적용하여 압축속도를 알고리즘[8]에 비하여 약 28배 개선하지만 본 논문에서 제안하는 방법보다 100배 이상의 시간을 요구하고 있다.

위의 방법들은 원자블록의 크기를 2×2 보다 크게 하고 화질을 높이기 위해서 탐색 점 및 유사변환 종류를 증가시키는데 이에 따라 압축시간이 크게 증가하게 된다. 또한 치역의 크기가 크면 압축률은 높일 수 있으나 화질을 높이는데 비효율적이게 된다. 본 논문에서 제안하는 방법은 원자블록의 크기를 2×2 로 하고 탐색 점의 수와 유사변환의 종류를 극히 제한하여 압축시간을 획기적으로 개선하고(512×512 영상에서 평균 0.024초) 압축률은 공간예측에 의하여 알고리즘[4] 보다 약 20% 개선하는데 원자블록의 크기가 작으므로 유사변환을 극히 제한하여도 알고리즘[1] 보다 평균 1.0dB 높은 화질을 유지하게 된다.

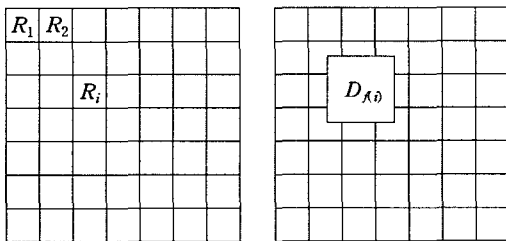


그림 1 프랙탈 코딩의 정의역과 치역

한 치역 R 의 화소(pixel) 수를 $|R|$ 이라 하고 압축에서 할당되는 비트(bit) 수를 $bit(R)$ 이라 하자. 치역 R 의 압축률은 $|R|/bit(R)$ 이므로 치역의 크기가 적어지면 압축률이 저하된다. 그런데 $|R|$ 이 아주 작으면 $bit(R)$ 을 구성하는 탐색영역, 유사변환의 종류 및 계수의 범위를 극히 제한하여도 오차가 적은 유사블록이 존재할 확률이 높아진다. 예를 들면, 512×512 크기의 영상에서 치역의 크기가 4×4 인 경우 고품질을 위해서는 보통 $bit(R) = 32 \pm \alpha$ (위치: 16~18 비트, $s=3 \sim 5$ 비트, $g=6 \sim 8$ 비트, rotation and flip: 3비트)[2]이 되어 약

4배의 압축이 된다. 치역의 크기가 2×2 인 경우 위와 같이 비트를 할당하면 압축이 거의 되지 않는다. 그러나 $bit(R)$ 을 치역의 크기가 4×4 인 경우보다 $1/4$ 이하로 사용하고 유사블록 오차를 줄일 수 있으면 더 좋은 화질/압축률을 얻을 수 있게 된다. 특히 탐색 점의 수가 현격히 적으므로 탐색시간이 현격히 줄어든다.

분할하는 치역 중 가장 작은 크기의 블록을 원자블록(atomic block)이라고 하자. 원자블록의 크기가 아주 작으면 일반적으로 인접한 블록간의 유사변환계수들에 좋은 상호관계(correlation)가 있게 되므로 인접한 블록들로부터 유사변환 계수를 예측하여 압축률을 향상시킬 수 있다. 위치 및 대비는 무손실 압축(lossless compression)을 하여야 하는데 오차와 변환에 사용되는 비트를 기준으로 공간예측(intra-prediction)을 하면 예측 성공률이 아주 높게 되어서 적은 비트를 사용하게 된다. 밝기가 가장 많은 비트를 사용하게 되는데 손실 압축(lossy compression)이 가능하고 극소 원자블록의 특성에 따라 이 값들은 영상을 가로 세로 각각 반으로 축소된 형태를 취하게 되므로 다양한 압축방법의 적용이 가능하다. 원자블록의 크기가 크면 변환계수간의 상호관계가 충분히 좋지 않으므로 위와 같은 공간예측을 적용하기가 어렵다. 따라서 기존의 프랙탈 코딩 방법에서는 변환계수 예측 보다는 블록을 합병하는 방법을 주로 사용하는데 본 방법은 기존의 블록합병에 의한 코딩 보다 512×512 크기의 영상에서 화질을 평균 1.1dB 향상시키고 압축시간은 평균 700 배 향상시킨다.

본 논문은 다음과 같이 구성된다. 2절에서는 극소 원자블록의 특성과 코딩에 대하여 논의한다. 3절에서는 변환예측 방법에 의한 압축방법을 논의한다. 4절에서는 PC(Intel Pentium IV 2.0G Hz, 512M DRAM)에서의 실험결과 및 분석이 주어진다. 5절에서는 결론이 주어진다.

2. 원자블록 코딩

픽셀의 수가 n 개 인 한 블록이 한계오차 δ 이내에서 한 블록과 유사 할 확률은 $(\delta/256)^n$ 에 비례하므로 블록의 크기가 커지거나 한계오차가 줄어들면 한계오차 이내의 유사블록을 찾을 확률이 급속히 떨어진다. 이와는 반대로 블록의 크기가 극히 작으면 지역성(locality)이 매우 강하게 되어 적은 탐색 점에서 단순한 변환을 하여도 유사블록과의 오차를 작게 할 수 있게 된다. 그러나 원자블록의 크기가 작을수록 압축률이 저하되므로 일반적인 프랙탈 압축방법은 화질/압축률을 높이기 위해서 원자블록의 크기를 크게 하고 탐색 점을 늘리거나 복잡한 변환을 사용하고 있다.

크기가 $m \times n$ 인 영상에서 $b \times b$ 치역에 대한 $2b \times 2b$ 정의역 전역탐색 점의 수는 $N=(m-2b+1)(n-2b+1)$ 가 되고 정의역의 위치를 저장하는데 $O(\log N)$ 비트를 할당하게 된다. 기존의 프랙탈 영상압축 방법에서는 대부분 원자블록의 크기를 4×4 로 하고 있다. 원자블록의 크기가 4×4 인 경우 고화질을 위해서는 탐색영역을 아주 크게 하여야 하며 압축시간은 탐색영역의 크기와 비례하므로 압축시간이 매우 길어지게 된다. 영상의 크기가 512×512 일 때 원자블록의 크기를 4×4 로 하고 전역탐색을 하면 블록 당 탐색 점의 수가 505×505 개가 되고 정의역 위치에 $9/8$ bpp(bit per pixel)를 사용하게 된다.

본 방법에서는 원자블록의 크기를 2×2 로 하고 기본적으로 4개의 고정된 탐색 점에 대하여 유사블록 탐색을 한다. 이 경우, 원자 블록 당 탐색 점의 수는 전역탐색에 비하여 $4/(m-7)(n-7)$ 가 되고 $1/2$ bpp를 사용하게 된다. 블록의 위치는 좌상 꼭짓점의 좌표로 나타내기로 하자. 한 치역 (x, y) 에 대한 4개의 정의역은 (x, y) , $(x-2, y)$, $(x, y-2)$, $(x-2, y-2)$ 으로 한다 (그림 2 참조). 각 정의역은 내부의 한 사분면에 치역을 포함하게 된다. $(x-1, y-1)$ 에 위치한 정의역은 복원(decode)시 수렴(convergency)에 문제가 있으므로 정의역에서 제외한다.

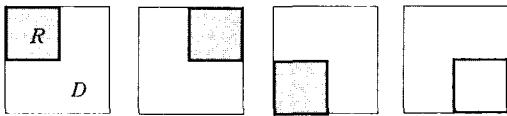


그림 2 4개의 정의역 위치

유사변환의 대비 값은 0.75와 0.5의 두 상수를 사용한다. 대비 값에 할당되는 bpp는 $1/4$ 이므로 원자블록의 크기를 4×4 로 할 때(보통 대비 값을 $4 \sim 32$ 개로 하고 $2 \sim 5$ 비트를 할당)와 거의 같게 된다. 대부분의 영상에서 대비 값 $s=0.5$ 는 에지(edge) 부근에 적용되고 $s=0.75$ 는 영상이 비교적 복잡하지 않은 부분에 적용된다.

밝기 값 g 는 양자화 하여 4비트를 할당한다. 식 (1)로부터 $g=R-s \cdot D$ 이 되는데 $s=0.5$ 일 때의 g 값이 평균적으로 $s=0.75$ 일 때의 g 값의 약 2배가 되므로 예측을 위한 상호관계 유지를 위하여 $s=0.75$ 이면 $2g$ 를 양자화 하여 저장한다.

본 방법은 하나의 4×4 치역 코딩을 4개의 2×2 서브블록으로 분할하여 각 서브블록이 독립적으로 극히 제한된 탐색범위와 유사변환으로 코딩 하는 것으로도

볼 수 있다. 2×2 치역 당 할당하는 총 비트 수는 $7/4$ bpp가 되므로 원자블록의 크기를 4×4 로 할 때의 2 bpp 보다 더 높은 압축률을 가진다. 유사블록 탐색시간은 전역탐색 시간의 약 $4/mn$ 이 된다. 기존의 프랙탈 압축 방법은 영상의 크기와 화질에 비례하여 블록 당 탐색 점의 수가 급속히 증가 하는 반면 본 방법은 블록 당 최대 탐색 점의 수가 일정 하므로 압축시간은 큰 영상에서 더 많은 향상을 보이고 영상의 특징에는 크게 영향을 받지 않는다. 대부분의 프랙탈 영상압축 방법에서 4×4 치역 코딩을 본 방법에서와 같은 4개의 2×2 치역 코딩으로 대체하면 압축시간을 현격히 줄이고 압축률과 화질을 향상시킬 수 있다.

3. 공간예측에 의한 프랙탈 코딩

영상을 원자블록들로 분할하여 2절에서와 같이 코딩을 하면 엔트로피(entropy) 코딩 이전에 $7/4$ bpp가 된다. 그러나 원자블록의 크기가 아주 작을 때는 변환계수간에 좋은 상호관계(correlation)가 존재하므로 예측(prediction) 및 엔트로피 코딩으로 압축률을 향상시킬 수 있다. i 번째 치역 R_i 와 유사변환 오차가 가장 적은 정의역의 위치, 대비, 밝기를 각각 $L_{\min}(i)$, $S_{\min}(i)$, $G_{\min}(i)$ 으로 나타내기로 하고 예측된 위치, 대비, 밝기를 각각 $L_p(i)$, $S_p(i)$, $G_p(i)$ 으로 나타내기로 하자. 먼저 알고리즘의 기본구조를 보면 예측된 위치 $L_p(i)$ 의 정의역이 예측된 대비 값 $S_p(i)$ 으로 한계오차 이내의 변환이 되면 밝기의 차이 값 $G_{\min}(i) - G_p(i)$ 만 저장한다. 이 경우를 복사(copy)라 한다. 복사가 아닌 경우 4개의 위치에서 최소오차 블록을 찾고 $L_{\min}(i)$, $S_{\min}(i)$ 와 $G_{\min}(i) - G_p(i)$ 를 저장한다.

위치 및 대비는 무 손실 압축이 되어야 하는데 예측은 유사블록탐색이 수행되기 전에 그림 3과 같이 이미 코딩된 치역 중에서 R_i 과 인접한 3개의 치역 R_a , R_b , R_c 을 참조한다. R_a 는 R_c 가 없을 경우에 참조된다. 예측 값은 중간 값(median)을 취하며 식 (3)과 식 (4)로 나타난다. 참조되는 블록이 영상의 경계(boundary)를 벗어나면 L 은 영상의 내부방향으로 값을 주고 S 는 0.5로 한다.

$$L_p(i) = \text{mid}(L(a), L(b), L(c)) \tag{3}$$

$$S_p(i) = \text{mid}(S(a), S(b), S(c)) \tag{4}$$

위치 및 대비는 무 손실(lossless) 압축을 하여야 하는 반면에 밝기는 손실(lossy) 압축이 가능하다. 한 정의역은 그림 3에서와 같이 치역 R_m 과 인접한 3개의 치역으로 구성되는 것으로 생각 할 수 있다(예를 들면,

R_d	R_b	R_c
R_a	R_i	R_e
R_f	R_g	R_h

그림 3 g 의 참조블록

R_d, R_b, R_a, R_m). 밝기는 한계오차이내 또는 최소오차로 변환되는 정의역 $D_{\lambda,0}$ 이 결정된 다음에 예측을 한다. 따라서 $D_{\lambda,0}$ 에 포함된 3개의 인접한 지역이 결정되는데 $D_{\lambda,0}$ 는 한계오차이내 또는 최소오차로 R_m 으로 변환이 되었으므로 이 지역들은 R_i 과 유사할 가능성이 높게 된다. 따라서 이들 지역 중에서 이미 코딩된 지역이 두개 이상이면 이들을 참조하고, 이미 코딩된 지역이 하나 이하이면 정의역에 포함되고 관계없이 이미 코딩된 4개의 인접한 블록 R_a, R_b, R_c, R_d 을 참조하여 모드를 결정한다. 전자의 경우 이들 블록 g 의 평균값, 후자의 경우 4개의 인접한 블록의 g 중 가장 유사한 값을 예측 값으로 한다. 여기서, 모드를 그대로 저장하면 많은 비트가 할당되므로 R_b 와 R_a 의 모드에 따라 확률적으로 높은 R_i 의 모드 테이블을 만들어서 그 값을 저장한다. 밝기 값에 양자화를 지나치게 하면 화질이 급속히 떨어지므로 고압축(low bit rate)에서는 밝기 값을 $m/2 \times n/2$ 크기의 이미지로 만든 다음 이를 다른 DCT, wavelet 등의 다른 영상압축 방법을 적용한다. 코딩은 각 지역에 대해서 위치와 대비를 먼저 예측하고 한계오차 이내에서 유사변환이 되면 모드를 copy로 한다. 예측된 위치와 대비에서 한계오차 이내에 유사변환이 되지 않으면 최소오차를 가지는 정의역을 탐색하고 모드를 non-copy로 한다. 여기서 non-copy 모드의 오차가 copy 모드의 오차 + ϵ 보다 크면 copy 모드로 환원시킨다(ϵ 값은 한계오차 값에 따라 결정된다). 원자블록의 크기를 2×2 로 하고 변환계수를 예측하는 프랙탈 코딩방법 QuickFractal은 그림 5와 같다.

Copy는 영상이 단순한 부분, non-copy는 영상이 복잡한 부분에 주로 위치한다. Copy에서는 $L(m), S(m)$ 을 저장하지 않으며 $G(m) - G_p(m)$ 이 0에 가까운 값이 많으므로 non-copy 보다 훨씬 적은 비트를 사용하게 된다. 따라서 압축률은 copy의 비율에 따라 결정된다. 실험결과 본 방법의 예측정확도가 아주 높아서 대부분의 영상에서 에지의 일부만이 non-copy가 되고 나머지는 copy가 되었다(그림 4참조). 512×512 Lenna와 Boat 영상의 copy 모드분포는 Lenna는 copy 93%,

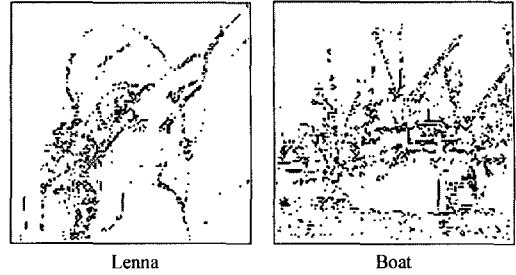


그림 4 copy와 non-copy 모드 분포(□: copy, ■: non-copy)

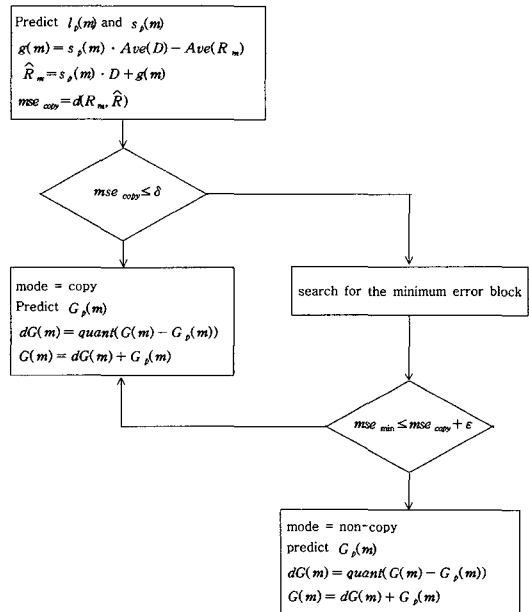


그림 5 QuickFractal

Boat는 copy 83%가 되었다. 연산 $G(m) = dG(m) + G_p(m)$ 은 차이 값의 양자화 따른 오차축적을 막기 위해서 필요하다.

4. 실험결과

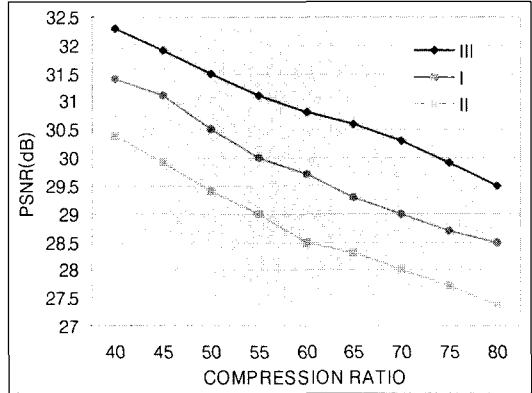
실험은 PC(Intel Pentium IV 1.6G Hz, 512M DRAM)에서 512×512 크기의 표준영상 Lenna, Baboon, Boat, Pepper에 대해서 수행되었다. Entropy 코딩은 context based arithmetic coding을 사용하였다.

QuickFractal를 Saupe[2]의 Horizontal-Vertical 분할방법과 Tong와 Wong[1]의 Adaptive Approximate Nearest Search 방법과 비교하여 보면 표 1과 같이 Adaptive Approximate Nearest Search 방법 보다 평균 압축시간이 약 700 배 향상되었다. 압축률은 copy/

non-copy에 비해하고 압축시간은 copy/non-copy가 감소하면 다소 길어지나 큰 차이는 없다. 화질은 10~20% 높은 압축률에서 PSNR이 평균 1.0dB 향상되었다. 특히, 원자블록의 크기가 다른 방법들에 비하여 극히 작으므로 에지부분이 보다 선명하게 나타난다. 원자블록의 크기를 크게 하면 변환계수간의 상호관계가 저하되고 예측의 정확도가 떨어져서 화질이 저하되면서 압축률 또한 저하된다. 따라서 copy 모드를 결정하는 한계오차 값을 조정하여 화질/압축률을 조절하는 것이 블록의 크기를 크게 하는 것 보다 효율적이다. 본 방법에서 비트를 가장 많이(평균 80% 이상) 사용하는 부분은 밝기 이므로 이를 효율적으로 압축하면 압축률이 더욱 향상될 수 있다.

Quick Fractal은 극히 작은 블록 단위로 코딩을 하기 때문에 에지와 무늬(texture)가 보다 선명하고 ringing, blocking artifact가 적어서 PSNR 대비 가시적인 화질이 다른 방법들에 비해서 향상 된다(그림 6 참조). Quick Fractal의 가장 큰 장점은 압축시간을 획기적으로 향상 시키는데 있다. 실험결과 CIF 크기의 영상에서 압축/복원 시간이 Quick Fractal은 평균 0.018초 이고 JPEG-2000은 0.061초가 되어서 JPEG-2000에 비하여 약 3배 빠른 것으로 나타났다. Quick Fractal은 기존의 프랙탈 코딩 방법에 비하여 화질, 압축률, 압축시간이 모두 우수한 반면에 JPEG-2000에 비하여 화질/압축률이 다소 저하되지만 압축시간이 빠르므로 저가의 시스

표 2 Quick Fractal과 블록병합방법과 Quadtree 방법과의 성능 비교 (I: 블록병합, II: Quadtree, III: Quick Fractal)



JPEG 2000, PSNR 33.75 Quick Fractal: PSNR 33.74
그림 6

표 1 Quick Fractal과 Horizontal-Vertical 분할방법과 Adaptive Approximate Nearest Search 방법과의 성능 비교

영상 (512x512)	HV			Adaptive			Quick Fractal		
	시간	PSNR	압축률	시간	PSNR	압축률	시간	PSNR	압축률
Lenna	60	35.80	8.31	17	35.83	9.04	0.022	36.25	11.05
Baboon	168	26.69	4.16	25	26.73	5.21	0.027	27.76	6.79
Boat	90	32.15	5.98	22	32.47	6.46	0.025	33.29	9.38
Pepper	60	33.17	8.12	16	33.47	9.15	0.021	35.07	11.95

영상 (256x256)	HV			Adaptive			Quick Fractal		
	시간	PSNR	압축률	시간	PSNR	압축률	시간	PSNR	압축률
Lenna	16	32.11	7.31	4	32.13	8.04	0.012	32.39	10.46
Baboon	54	26.01	4.01	6	26.12	4.56	0.014	27.10	6.16
Boat	27	30.14	5.22	5	30.42	6.15	0.013	31.02	7.97
Pepper	16	31.32	7.18	4	31.49	8.02	0.012	32.06	10.42

영상 (325x288)	HV			Adaptive			Quick Fractal		
	시간	PSNR	압축률	시간	PSNR	압축률	시간	PSNR	압축률
Akiyo	13	35.05	7.31	3	35.66	15.44	0.016	36.17	21.04
News	21	32.21	4.01	4	32.97	11.61	0.017	33.89	15.66
Foreman	22	33.26	5.22	4	33.02	11.01	0.016	33.54	15.76
Stepan	28	30.07	7.18	5	30.24	7.34	0.017	31.37	10.42

템에서 구현되는 동영상에서는 JPEG-2000에 비하여 유리 할 수 있다.

5. 결론

본 논문은 극소 원자블록에 공간예측을 적용하여 프랙탈 영상압축의 압축시간을 향상시키고 화질/압축률을 향상시키는 방법을 제안하였다. 특히, 본 방법은 탐색범위를 극히 제한하기 때문에 기존의 프랙탈 압축방법들보다 압축시간을 획기적으로 향상시킨다. 또한, 변환계수들이 좋은 상호관계를 유지함을 이용하여 변환계수 예측으로 화질/압축률을 향상시킨다. 여기서, 화질/압축률은 예측의 성공률에 비례하는데 주변블록의 정보를 보다 더 잘 활용하면 예측의 성공률이 더욱 높아질 수 있다. 본 방법에서 비트를 가장 많이 사용하는 부분은 밝기 이므로 이를 보다 효율적으로 압축하면 압축률이 더욱 향상 될 수 있다.

참 고 문 헌

- [1] Tong, C.S. and Wong, M., "Adaptive Approximate Nearest Neighbor Search for Fractal Image Compression," IEEE Trans. on Image Processing, Vol. 11, No. 6, pp. 605-615, 2002.
- [2] Saupe, D., "Fractal Image Compression via Nearest Neighbor Search," in Conf. Proc. NATO ASI Fractal Image Encoding and Analysis, Trondheim, Norway, pp. 1-23, 1995.
- [3] Ruhl, M., Hartenstein, H. and Saupe, D., "Adaptive Partitioning for Fractal Image Compression," IEEE International Conference of Image Processing, pp. 21-24, 1997.
- [4] Hartenstein, H., Ruhl, M. and Saupe, D., "Region-Based Fractal Image Compression," IEEE Trans. on Image Processing, Vol. 9, No.7, pp. 1171-1184, 2000.
- [5] Belloulata, K. and Konard, J., "Fractal Image Compression With Region-Based Functionality," IEEE Trans. on Image Processing, Vol. 11, No. 4, pp. 351-362, 2002.
- [6] Lai, C., Lam, K. and Siu, W., "A Fast Fractal Image Coding Based on Kick-Out and Zero Contrast Conditions," IEEE Trans. on Image Processing, Vol. 12, No.11, pp. 1398~1403, 2003.
- [7] Cardinal, J., "Fast Fractal Compression of Grey-scale Images," IEEE Trans. on Image Processing, Vol. 10, No.1, pp. 159-163, January 2001.
- [8] Lee, S., "A Fast Variance-Ordered Domain Block Search Algorithm for Fractal Encoding," IEEE Transactions on Consumer Electronics, Vol. 45, No. 2, pp. 275-277, May 1999.
- [9] Lai, C., Lam, K. and Siu, W., "Improved Searching Scheme for Fractal Image Coding," Electronics

Letters, Vol. 38, No. 25, pp. 1653-1654, 2002.

- [10] He, C., Yang, S.X. and Huang, X., "Variance-based Accelerating Scheme for Fractal Image Encoding," Electronics Letters, Vol. 40, No. 2, pp. 115-116, 2004.



위 영 철

학사 SUNY at Albany 전산학과(1982).

석사 SUNY at Albany 전산학과(1984).

박사 SUNY at Albany 전산학과(1989).

1990년~1995년 삼성종합기술원 수석연구원.

1995년~1998년 현대전자 정보시스템사업본부 부장.

1998년~현재 아주대

학교 정보통신 대학 조교수. 관심분야는 영상압축, 컴퓨터 그래픽스 및 알고리즘