

# SAS 기법을 이용한 웨이브릿 변환 기반 프랙탈 영상 압축

°정 태일\*, 강 경원\*, 문 광석\*, 권 기룡\*\*\*, 류 권열\*\*

\* 부경대학교 전자 컴퓨터 정보통신공학부

\*\* 위덕대학교 멀티미디어 공학과

\*\*\* 부산외국어대학교 전자 컴퓨터공학부

## The Fractal Image Compression Based on the Wavelet Transform Using the SAS Techniques

°Tae-il Jeong\*, Kyung-won Kang\*, Kwang-seok Moon\*  
Ki-ryong Kwon\*\*\*, Kwon-yeol Ryu\*\*

\* School of Elec., Computer and Telematics Eng., Pukyong National Univ.

\*\*\* School of Elec. and Computer Eng., Pusan Univ. of Foreign Studies

\*\* Dept. of Multimedia Engineering, Uiduk University

e-mail : tijeong@hanmail.net

### 요 약

기존의 웨이브릿 기반 프랙탈 압축 방법은 전 영역에 대하여 최적의 정의역을 탐색하므로, 부호화 과정에서 많은 탐색시간이 소요되는 단점이 있다. 그래서 본 논문에서는 웨이브릿 변환영역에서 SAS(Self Affine System) 기법을 이용한 웨이브릿 변환 기반 프랙탈 영상 압축 방법을 제안한다. 웨이브릿 변환영역에서 정의역과 치역을 구성하고, 각각의 정의역과 치역에 대해 모든 블록을 탐색하는 것이 아니라, 각 대역별로 공간적으로 같은 위치에 있는 블록을 정의역으로 선택한다. 이와 같이 웨이브릿 변환 영역에서 정의역 탐색과정이 필요 없는 SAS 프랙탈 영상 압축 방법을 도입하여 부호화 과정에서 곱셈 계산량을 감소시켜 고속 부호화를 가능하게 하였고, 복호화 과정에서 각 레벨별로 서로 다른 스케일을 사용하여 화질을 개선하였다.

### I. 서 론

멀티미디어 정보 중에서 전달 효과가 가장 큰 영상정보는 정보량이 방대하므로, 이를 제한된 전송채널이나 저장매체에서 효율적으로 활용하기 위해서 영상 데이터 압축은 필수적이다. DCT(Discrete Cosine Transform)를 근간으로

하는 정지영상 압축 표준인 JPEG(Joint Photographic Experts Group)은 압축률이 높을 경우, 블록킹 현상등으로 영상의 화질이 떨어지는 단점이 있다. DCT의 단점을 해결하기 위해, 이산 웨이브릿 변환(Discrete Wavelet Transform : DWT) 영역에서의 영상압축 방법이 제안되었다[1,2]. 이 방법은 기저벡터로 분할한 다해상도 영상에서 부호화 함으로서 높은 압축률에서도 복원된 영상의 블록킹 현상(blocking effect)을 제거시킨다.

프랙탈 영상압축은 원 영상을 치역 블록과 정의역 블록으로 분할한다. 그래서 이 블록을 가장 잘 근사화 하는 정의역 블록의 축소변환 계수를 부호화 한다[3,4]. 이러한 부호화 과정에서 치역 블록내의 계수들을 가장 잘 근사화하는 정의역 블록을 찾는 탐색과정에서 많은 시간이 필요하여, 실시간 부호화가 어려우며, 블록기반으로 부호화하므로 높은 압축률에서 블록킹 현상이 발생한다. 프랙탈 부호화 방법 중에서 SAS 기법은 블록 단위의 부호화 알고리즘으로서, 정의역 탐색과정이 없기 때문에 다른 프랙탈 부호화 방법에 비해 부호화 속도가 빠르다[5].

시각에 민감한 에지 성분을 잘 복원하고 블록킹 현상이 감소하는 웨이브릿과 프랙탈의 장점을

결합한 웨이브릿 기반의 프랙탈 부호화 방법이 연구되고 있다. 이 방법은 DWT 영역에서 서로 다른 고주파 부대역간에 유사성을 이용하여 프랙탈 부호화 한다. Krupnik 등<sup>[6]</sup>이 제안한 웨이브릿 기반 프랙탈 영상압축 방법들은 부호화과정에서 최적의 정의역을 찾기 위하여 탐색과정이 필요하므로 부호화 과정에 많은 시간이 소요된다.

본 논문에서는 부호화 과정에서 탐색과정에 소요되는 시간을 개선하기 위해, 웨이브릿 변환영역에서 SAS 기법을 이용한 웨이브릿 변환 기반 프랙탈 영상 압축 방법을 제안한다. 웨이브릿 변환영역에서 정의역과 치역을 구성하고, 각각의 정의역과 치역에 대해 모든 블록을 탐색하는 것이 아니라, 각 대역별로 공간적으로 같은 위치에 있는 블록을 정의역으로 선택한다. 이와 같이 SAS 프랙탈 영상 압축 방법을 도입하여, 부호화 과정에서 곱셈 계산량을 감소시켜 고속 부호화를 가능하게 하고, 복호화 과정에서 세 개의 스케일을 사용하여 SAS 기법의 단점인 화질 개선이 가능한 알고리즘을 제안한다.

## II. 이론적 배경

### 2.1 웨이브릿 변환

웨이브릿 변환에서의 기저함수들은 원형 웨이브릿을 천이(translation) 및 확장/수축(dilation)을 시킴으로서 만들어진다. 이에 대한 웨이브릿 기저함수들의 선형결합은

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right), \quad a, b \in \mathbb{R}, a \neq 0 \quad (1)$$

이다. 여기서  $a$ 는 원형 웨이브릿을 확장/수축시키는 스케일 변수이고,  $b$ 는 이동을 나타내는 천이변수이다. 어떤 신호  $f(t)$ 에 대한 웨이브릿 변환은

$$Wf(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \psi\left(\frac{t-b}{a}\right) dx \quad (2)$$

로 정의한다. 여기서 매개변수  $a, b$ 가 정수일 때 이산 웨이브릿 변환이라고 한다.

### 2.2 SAS 프랙탈 영상 압축

Monro 등<sup>[5]</sup>이 제안한 SAS 부호화 기법은 정의역 탐색범위를 치역의 인접 블록으로만 제한하여 부호화 속도를 높이는 방법이다. 즉 4개의 치역 블록을 1개의 정의역에 대해 탐색과정 없이 정의역으로 선택한다. 이 방법의 단점은 최적의 정의역을 선택할 수 없으므로 복원화질이 떨어진

다.

### 2.3 기존의 웨이브릿 변환 기반 프랙탈 영상압축

Krupnik 등은 공간영역에서 동일 위치에 대응되는 서로 다른 부대역간의 DWT 계수들을 묶은 블록들로서 치역 블록과 정의역 블록으로 구분하였다. 각각의 치역 블록과 정의역 블록에 대해 프랙탈 부호화함으로써 DWT 영역에서 자기 유사성을 찾는 방법을 제안하였다.

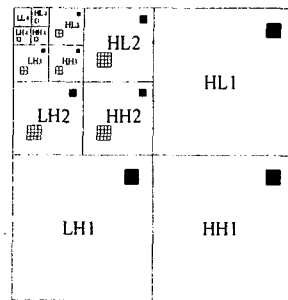


그림 1. 웨이브릿 기반 프랙탈 영상 압축(분할횟수=4, B=8)

그림 1은 4회 분할했을 때, 공간영역에서 정의역 블록(□)과 치역 블록(■)을 나타내었다. 정의역 블록은 LL4를 제외한 LH4, HH4, HL4에서부터 정의역 블록을 계층적으로 구성하게 되고, 치역 블록은 LH3, HH3, HL3에서부터 같은 방법으로 구성된다. 여기서 L은  $L = \log_2(B)$ 이다. B는 정의역 블록과 치역 블록의 크기이며, 이들 블록들은 공간적으로 같은 위치에 존재하고, 이 AC계수들을 묶은 블록(8×8)을 그림 2에 나타내었다.

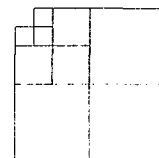


그림 2. DWT영역에서 정의역 및 치역블럭

DWT 영역에서 치역과 정의역 블록의 AC 성분에 해당하는 고주파 부대역의 DWT 계수들에 대한 프랙탈 코드는 DWT 영역에서 원래의 신호  $R_i^H$ 와 근사화된 신호  $\hat{R}_i^H$ 간의 자승 오차는

$$d^2(\hat{R}_i^H, R_i^H) = \sum_{p=2}^B (R_i^p - s \cdot D_j^p)^2 \quad (3)$$

을 최소화하는 스케일 인자  $s$ 와 정의역 블록의 위치 정보이다.

### III. 제안한 방법

본 논문에서는 웨이브릿 변환 상태에서 동일한 공간 위치에 대응되는 서로 다른 부대역간의 DWT 계수들을 묶은 다음, 치역 블록과 정의역 블록을 구성한다. 각각의 치역 블록과 정의역 블록에 대하여 모든 정의역 블록을 탐색하는 것이 아니라, SAS 기법을 이용하여 정의역 블록 탐색 없이, 각 대역별로 공간적으로 같은 위치에 있는 블록을 정의역으로 선택한다. 그래서 부호화 과정에서 곱셈 계산량을 감소시켜 고속 부호화를 가능하게 하고, 복호화 과정에서 각 레벨별로 서로 다른 스케일을 사용하여 화질이 개선되는 알고리즘을 제안한다.

제안한 방법의 부호화 과정을 그림 3 (a)에 나타내었다. 정의역 블록과 치역 블록은 공간적으로 같은 위치에 존재하므로,  $B \times B$  블록 단위로 4개의 치역 블록이 축소된 정의역 블록 1개에 선택되어  $s$ 를 구함으로써 부호화가 이루어진다(그림 3에서 HH1의 치역과 HH2의 정의역- $B \times B$  블록 단위이지만, 큰 그림 영역에서 설명하기 위해, HH1과 HH2 레벨에 나타내었음). SAS 기법의 단점인 화질을 높이기 위하여 제안한 방법은 두 가지 방법을 사용하였다. 첫 번째 방법은 DWT 계수에 절대값을 취한 후,  $B \times B$  블록 단위로  $s$ 값을 구하는 방법이다.

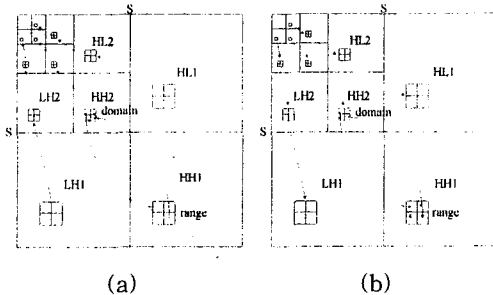


그림 3. 제안한 방법 1의 부호화 및 복호화 과정 (a)부호화 (b)복호화

제안한 방법의 첫 번째 복호화 과정은 정의역과 치역이 공간적으로 같은 위치에 있으므로, 정의역 정보로부터 하위의 치역 블록들을 계층적으로 복원이 가능하다. 즉, 그림 3(b)에서처럼 최상위 정의역(LH4, HH4, HL4)에서 최상위 치역(LH3, HH3, HL3)을 계층적으로 복원이 가능하고, 이때 적용된  $s$ 는 모두 같은  $s$ 이다.

두 번째 방법은 첫 번째 방법에서처럼  $B \times B$  블록 단위로  $s$ 를 구하는 것이 아니라, 레벨별로 프랙탈 부호화 한다. 즉, 그림 4(a)에서 치역

(LH3, HH3, HL3)에 대한 정의역은 공간적으로 같은 위치의 상위 레벨인 LH4, HH4, HL4블록이고, 이들 각각의 정의역 및 치역 블록에 대해 첫 번째 스케일 인자  $s_1$ 을 구한다. 같은 방법으로 다음 하위의 레벨에 대해 정의역과 치역 사이의 두 번째, 세 번째 스케일 인자  $s_2, s_3$ 를 구한다. 이러한 구현의 목적은 일반적으로 DWT 영역에서 AC계수는 저주파 영역으로 갈수록 값이 크고, 고주파 영역으로 갈수록 값이 작기 때문에, 저주파 영역에서의 오차가 고주파 영역으로 누적되는 누적오차를 감소하기 위함이다.

제안한 방법의 두 번째 복호화 과정은 그림 4(b)에 나타내었고, 첫 번째 방법과 동일하며, 차이점은 저주파 영역에서 고주파 영역으로의 누적 오차를 줄이기 위해 레벨별로 서로 다른  $s$ 를 적용하는 것이 다르다. 즉, 레벨 3의 치역 블록을 복호화 하기 위해  $s_1$ 을 적용하여 복원하고, 레벨 2는  $s_2$ , 레벨 1은  $s_3$ 을 적용하여 복원한다.

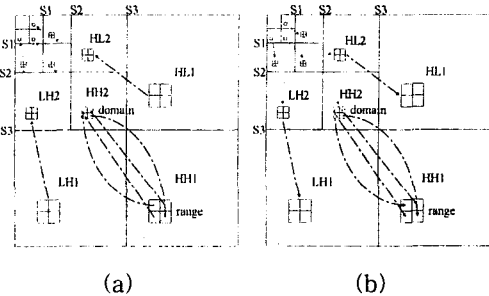


그림 4. 제안한 방법 2의 부호화 및 복호화 과정 (a)부호화 (b)복호화

### IV. 모의실험 및 결과

실험에 사용한 영상은  $512 \times 512$  8bit를 갖는 표준 영상들을 사용하였고, 9/7 쌍직교 웨이브릿을 사용하였다. LL4 대역(그림 2에서)은 7bit/coeff, LH4, HH4, HL4 대역은 6bit/coeff, scaling factor  $s$ 는 6bit로 모두 uniform 양자화 하였다.

표 1은  $515 \times 512$  Lena 이미지에 대하여 분할 횟수에 따른 기존의 방법과 제안한 방법들의 화소당 곱셈 계산량, PSNR, 압축률 비교를 나타내었다. 제안한 방법들의 곱셈 계산량이 크게 감소하였고, 제안한 방법 2에서처럼 각 레벨별로 프랙탈 압축함으로서, 큰 AC계수를 갖는 저주파 영역의 오차가 작은 계수를 갖는 고주파 영역으로 누적되는 누적오차를 줄일 수 있다. 또, 제안

표 1. 분할횟수에 따른 곱셈 계산량, PSNR, 압축률 비교(Lena 이미지)

분할횟수	기존의 방법			제안한 방법 1			제안한 방법 2		
	곱셈계산량 [Mul./pel]	PSNR [dB]	압축률 [bpp]	곱셈계산량 [Mul./pel]	PSNR [dB]	압축률 [bpp]	곱셈계산량 [Mul./pel]	PSNR [dB]	압축률 [bpp]
3	4352.25	39.14	1.52	1.31	33.76	0.77	1.44	35.50	1.14
4	1042.25	32.30	0.35	1.27	29.27	0.19	1.30	31.05	0.38
5	257.25	27.50	0.08	1.25	26.05	0.05	1.26	27.58	0.12

한 방법 1의 압축률(bpp)이 기존의 방법보다 낮은 이유는 제안한 방법에서는 정의역과 치역 블록이 공간적으로 같은 위치에 존재하므로, 기존의 방법처럼 별도의 정의역 정보가 필요 없기 때문이다.

그림 5는 분할횟수가 4회일 때, 각각 기존의 방법과 제안한 방법 2에 의해 복원한 영상을 나타낸다. 기존의 방법은 PSNR이 32.30[dB]이었으며, 제안한 방법 2의 PSNR은 31.05[dB]로서 대략 1.2[dB] 감소하였으나, 부호화 과정에서 곱셈 계산량이 기존의 방법은 화소당 1042.25회, 제안한 방법 2는 1.30회로 크게 개선되었다.



(a) 기존의 방법 (b) 제안한 방법 2  
그림 5. 복원영상들 (a) PSNR=32.30[dB]  
(b) PSNR=31.05[dB]

표 2는 분할 횟수가 4회일 때, 다른 512×512 표준 영상에 대하여 기존의 방법과 제안한 방법들의 PSNR 및 부호화시의 화소당 곱셈 계산량을 비교하였다.

표 2. 영상에 따른 PSNR, 곱셈 계산량 비교 (4회 분할)

영상	기존의 방법		방법 1		방법 2	
	PSNR [dB]	계산량	PSNR [dB]	계산량	PSNR [dB]	계산량
Boat	29.47	1042.3	26.10	1.27	27.79	1.30
Bridge	24.35	1042.3	22.78	1.27	24.17	1.30
Pepper	31.29	1042.3	27.69	1.27	29.20	1.30
Goldhill	29.29	1042.3	27.61	1.27	28.95	1.30

전체적으로 표 1에서처럼 제안한 방법들이 기존의 방법보다 곱셈 계산량이 크게 개선되었다.

### V. 결론

기존의 웨이브릿 기반 프랙탈 영상 압축은 부호화 과정에서 각각의 치역에 대해 최적의 정의역을 찾기 때문에 많은 시간이 소요되었다. 그래서 본 논문에서는 부호화 과정에서 정의역 탐색 시간이 필요 없는 SAS 기법을 이용한 웨이브릿 기반 프랙탈 영상 압축 방법을 제안하였다. 분할 횟수가 4회일 때, 부호화시 화소당 곱셈 계산량이 기존의 방법은 1042.25회, 제안한 방법 1은 1.27회, 제안한 방법 2는 1.30회로서, 곱셈 계산량을 감소시켜 고속 부호화를 가능하게 하였다. 또 복호화 과정에서 각 레벨별로 서로 다른 스케일을 적용하여 화질을 개선하였다.

향후 연구과제로서는 SAS 기법의 단점인 화질 개선에 관한 연구가 있어야 될 것 같다.

### 참고 문헌

1. M. Antonini, M. Barlaud, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. on Image Processing*, vol. 1, no. 2, pp. 244-250, Apr. 1992.
2. S. G. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Trans. of PAMI*, vol. 11, no. 7, pp. 674-693, Jul. 1989.
3. A. E. Jacquin, "Image Coding Based on a Fractal Theory of Iterated Contractive Image Transforms," *IEEE Trans. on Image Processing*, vol. 1, no. 1, pp. 18-30, Jan. 1992.
4. Y. Fisher, *Fractal Image Compression: Theory and Application to Digital Images*, Springer Verlag, New York, 1994.
5. D. M. Monro and F. Dudbridge, "Fractal Approximation of Image Blocks," *IEEE ICASSP*, pp. III-485-488, 1992.
6. H. Krupnik, D. Mallah and E. Karnin, "Fractal representation of Image via the Discrete Wavelet transform," in *IEEE 18th Conv. of EE Israel*, in Tel-Aviv, Mar. 1995..