정규화된 분산을 이용한 프랙탈 압축방법

김 종 구†·함 도 용†·위 영 철††·김 하 진†††

요 약

프랙탈 코딩은 높은 압축률을 포함한 여러 가지 장점을 가지고 있으나 유사블록 탐색에 긴 시간이 소요되는 문제점을 가지고 있다. 본 논문은 각 블록의 정규화 된 분산 값은 명도(contrast)와 밝기(brightness)에 독립적임을 발견하고, 이를 이용하여 d(key의 수)차원 공간에서 최근접 부근탐색(nearest neighbor search)을 하여 효율적인 유사블록을 탐색하는 방법을 제안한다. 본 방법은 각 치역 블록 당 $O(\log N)$,(N: 정의역 블록 수) 시간에 유사 정의역 블록 찾을 수 있음을 보였다. 압축처리 된 이미지는 각 치역 블록 당 $O(N)$시간이 요구되는 전체탐색의 PSNR(Peak Signal Noise Ratio)과 거의 같은 값을 얻게 되었다. 또한, 본 방법은 에지가 많은 이미지에도 전체탐색과 거의 유사한 PSNR로 압축되는 장점을 가진다.

A Fast Fractal Image Compression Using The Normalized Variance

Jong Koo Kim†·Do Yong Hamn†·Young-Cheul Wee††·Ha-Jine Kimn†††

ABSTRACT

Fractal image coding suffers from the long search time of domain pool although it provides many promising properties including the high compression ratio. We find that the normalized variance of a block is independent of contrast, brightness. Using this observation, we introduce a self similar block searching method employing the d-dimensional nearest neighbor searching. This method takes $O(\log N)$ time for searching the self similar domain blocks for each range block where N is the number of domain blocks. PSNR (Peak Signal Noise Ratio) of this method is similar to that of the full search method that requires $O(N)$ time for each range block. Moreover, the image quality of this method is independent of the number of edges in the image.

키워드 : 정규화분산(Normalized Variance), 탐색키(Serach Key), 최근접탐색(Nearest Neighbor Search)

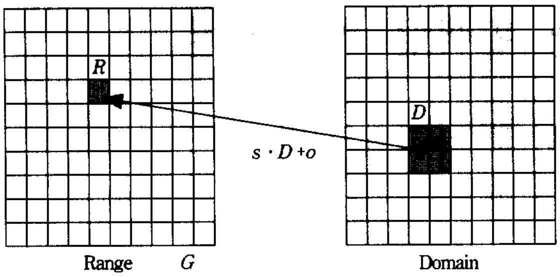
1. 서 론

프랙탈 이미지 압축은 압축할 이미지를 치역 블록들로 분할한 다음 각 치역블록과 유사한 부분 이미지(정의역 블록)의 위치, 명도(contrast), 밝기(brightness)를 저장하여 이미지를 압축하는 방법이다(그림 1). 두 블록 $A=(a_1, a_2, \cdots a_m)$와 $B=(b_1, b_2, \cdots, b_m)$간의 유사성은 RMSE(Root Mean Square Error)로 측정되며 $d_e(A,B)=\sqrt{\frac{1}{m}\sum_{1\le i\le m}(a_i-b_i)^2}$로 표현된다. 각 치역 블록마다 RMSE가 최소가 되는 정의역 블록을 찾기 위하여 전체탐색(모든 정의역 블록을 비교)을 할 경우 정의역 블록 수가 많기 때문에 $d_e(A,B)$ 계산에 많은 시간을 소모하게 된다. 이를 개선하는 여러 가지 방법[2-7]들이 제안되었으나 압축속도는 여전히 많은 개선이 필요하다.

한 치역 블록 R과 한 정의역 블록 D가 유사하다는 것은 D를 명도 s와 밝기 o로 변환한 $\hat{D}=s\cdot D+o$와 R간의 RMSE가 적다는 것을 의미한다. 한 치역 블록 R과 한 정의역 블록 D의 유사성은 R과 D간의 비교가 아니라 R과 $\hat{D}$ 간의 비교를 해야하며 $\hat{D}$는 변수 s와 o에 따라 변하기 때문에 효율적인 탐색구조를 만들기 위해서는 정의역 블록의 픽셀 값들을 직접적인 key로 사용할 수 없으며 s와 o에 독립적인 key 값들이 필요하게 된다.

본 논문에서는 정규화 된 분산 값이 s와 o에 독립적임을 발견하고 이를 키로 사용하여 d차원(키의 수)공간에서 최근접 부근 탐색을 하여 유사블록을 각 치역블록 당 $O(\log N)$시간에 정의역 블록을 탐색하는 방법을 제안한다. 본 방법은 부분블록 또는 픽셀 각각의 키 값을 만들기 때문에 벡터 변환에 의한 키 값을 사용하는 [1]의 경우 보다 더 나은 PSNR을 가지며 이미지의 유형에 관계없이 전체탐색과 유사한 PSNR을 가지게 된다.

† 정 회 원 : 아주대학교 대학원 컴퓨터공학과
†† 정 회 원 : 아주대학교 정보 및 컴퓨터공학부 조교수
††† 정 회 원 : 아주대학교 정보 및 컴퓨터공학부 교수
논문접수 : 2001년 2월 21일, 심사완료 : 2001년 11월 2일



(그림 1) 치역블록과의 비교를 위한 정의역 블록 변환

프랙탈 이미지 압축/복원에서 압축과정은 (그림 1)과 같이 전체 이미지 G를 겹치지 않는 치역(non-overlapping range) 블록들로 분할하고 각 치역 블록마다 유사한 정의역 블록의 위치, 명도, 밝기 정보를 저장하고 복원과정은 임의의 이미지로부터 압축과정에서 저장해 둔 위치, 명도, 밝기에 따른 변환을 반복적으로 수행하여 원래의 이미지와 유사한 이미지를 찾게 된다. 한 치역 블록 R이 한 정의역 블록 D와 유사하다는 것은 D를 명도 s와 밝기 o로 변환한 $\widehat{D}=s \cdot D+o$와 R간의 RMSE가 적다는 것을 의미한다. 즉, 한 치역블록 R과 가장 유사한 정의역 블록 D는 $d_e(A, \widehat{D})$가 최소가 되는 블록이 된다. 프랙탈 이미지 압축/복원에서 복원과정은 비교적 빠른 시간에 처리되나 압축과정에서 유사블록 탐색에 많은 시간이 소요되는 이유는 치역 블록 수가 K이고 정의역 블록수가 N일 때 유사블록 탐색에서 전체탐색을 할 경우 $O(K \cdot N)$의 RMSE 계산이 요구되기 때문이다.

[1]에서는 각 블록을 하나의 벡터로 하여 기저 벡터들과 내적 값을 계산하여 s와 o에 독립적인 key 값들을 구한 다음 최근접 부근 탐색을 하여 유사블록을 탐색하는 방법을 제안하였다. 최근접 부근 탐색 방법을 유사성 탐색에 도입하기 위해서는 R이 $\widehat{D}=s \cdot D+o$와 비교되고 s와 o는 변수이기 때문에 s와 o에 독립적인 키 값들을 만들어야 하며 키 값들의 유사성이 R과 $\widehat{D}$의 유사성과 가능한 일치하여야 한다. [1]의 방법은 최근접 부근 탐색을 도입하여 $O(K \cdot \log N)$의 빠른 시간에 유사블록 탐색을 처리하지만 이미지의 유형과 기저 벡터들의 선택에 따라 복원 이미지의 품질이 저하되는 단점을 가진다. 특히 에지가 많은 이미지를 잘 처리하지 못하는 경우가 많다. [3, 4]의 분산을 이용한 방법은 치역블록과 거리가 가장 가까운 정의역 블록을 찾기 위하여 전체 정의역 블록 집합을 탐색할 필요 없이 주어진 오차 척도의 한계를 만족하는 영역내의 정의역 블록 탐색만으로 전체 탐색 경우와 비교하여 최소한의 거리 안에 존재하는 정의역 블록들을 찾는 방법이다. 본 논문은 정규화된 분산 값에 대한 최근접 부근 탐색을 하여 유사블록 탐색을 $O(K \cdot \log N)$ 시간에 처리하는 방법을 제안하며, 탐색키를 여러 가지 패턴으로 생성하여 패턴은 다르나 분산 값이 같은 경우에도 효율적으로 유사한 정의역 블록을 탐색하도록 한다. 또한 부분블록에 대한 독립적인 키 값들을 사용하기 때문에 이미지의 유형에 관계없이 전체탐색과 유사한 PSNR을 가지며, 벡터 변환 방법에 비하여 키 값 계산이 간단하므로 탐색구조를 만들기 전의 전처리 시간이 단축된다.

2. Search Key

한 블록 $B=(b_1, b_2, \cdots, b_m)$에 있는 픽셀들의 평균값은 $Ave(B)=\frac{1}{m}\sum_{1 \le i \le m} b_i$이 되고 분산값은 $Var(B)=\sum_{1 \le i \le m}(b_i-Ave(B))^2$가 된다. 한 정의역 블록 $D=(d_1, d_2, \cdots, d_m)$가 변환된 $\widehat{D}=s \cdot D+o$ 의 분산은

$$Var(\widehat{D})=\sum_{1 \le i \le m}(s \cdot d_i+o-Ave(\widehat{D}))^2 \text{ 가 된다.} \quad (1)$$

여기서,

$$\begin{aligned} Ave(\widehat{D}) &= \frac{1}{m}\sum_{1 \le i \le m}(s \cdot d_i+o) \\ &= s \cdot Ave(D)+o \text{ 임으로} \end{aligned} \quad (2)$$

$$\begin{aligned} Var(\widehat{D}) &= \sum_{1 \le i \le m}(s \cdot d_i+o-Ave(\widehat{D}))^2 \\ &= \sum_{1 \le i \le m}(s \cdot d_i-s \cdot Ave(D))^2 \\ &= s^2 \cdot \sum_{1 \le i \le m}(d_i-Ave(D))^2=s^2 \cdot Var(D) \end{aligned} \quad (3)$$

가 된다. 따라서 $Var(\widehat{D})=s^2 \cdot Var(D)$는 변수 o에 독립적인 값이 된다.

한 정의역 블록 D의 부분블록 $P(D)$에 대하여 정규화 분산을 $P(D)=\frac{Var(P(D))}{Var(D)}$이라고 하자.

부분블록 $P(D)$의 분산은 위와 유사한 이유로 $Var(\widehat{P(D)})=s^2 \cdot Var(P(D))$가 된다. 따라서, 부분블록 $P(D)$의 정규화 된 분산은

$$\frac{Var(\widehat{P(D)})}{Var(D)}=\frac{s^2 \cdot Var(P(D))}{s^2 \cdot Var(D)}=\frac{Var(P(D))}{Var(D)} \quad (4)$$

이 되어 변수 s와 o에 독립적인 값이 되며 R과 유사한 정의역 블록을 찾는 키로 사용될 수 있다.

3. 알고리즘

한 치역 블록 R이 한 정의역 블록 D와 유사하면 정규화된 분산도 유사하게 된다. 그러나, 이의 역은 성립하지 않으므로 탐색된 최적의 정규화 된 분산이 R과 D의 유사성을 보장하지 못한다. 따라서, 최근접 부근 탐색시 k-nearest neighbor 탐색이 필요하게 된다. 본 논문의 알고리즘은 다

음과 같이 구성된다.

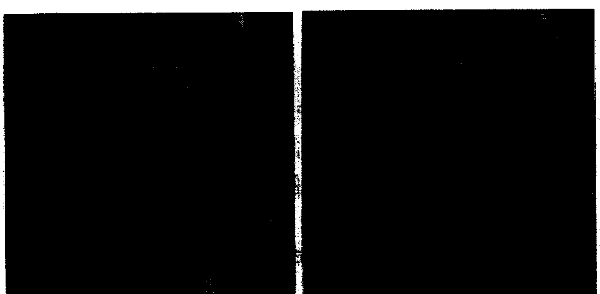
Fractal Compression
1. 정의역 블록들의 정규화 된 분산 키 값들을 계산 한다.
   1.1 치역블록과 여러 패턴의 정규화 된 분산 키 값들이 적용 된 정의역 블록과 값을 비교한다.
   1.2 두 블록간의 RMSE 차이가 적은 순서로 정렬한다.
2. 정규화 된 분산 키 값으로 최근접 부근 탐색구조를 구축한다.
3. 각 치역블록에 대하여 정규화 된 분산 키 값에 대한 가장 근접한 k개의 정의역 블록을 찾는다.
4. k개의 블록 중 RMSE가 가장 적은 블록을 찾는다.

End Fractal Compression

본 방법은 부분블록 각각의 키 값을 만들기 때문에 벡터 변환에 의한 키 값을 사용하는 [1]의 경우 보다 더 나은 PSNR을 가지며 이미지의 유형에 관계없이 전체탐색과 유사한 PSNR을 가지게 된다. 최근접 부근 탐색 알고리즘은 [8, 9], 등 여러 가지 알고리즘들이 알려져 있다. 이러한 알고리즘들은 각 range 블록 당 $O(\log N)$시간, $N=$ domain 블록의 수에 최근접 부근 탐색하므로 본 방법은 치역 블록 수가 K일 때 $O(K \cdot \log N)$의 시간에 유사블록 탐색을 처리하게된다.

4. 실험 및 검증

제안한 방법을 256×256 크기의 8 bit Gray Scale Lena, Mandrill 이미지에 적용하였다. 치역 블록의 크기를 4×4로 하고 정의역 블록의 크기는 8×8로 하였다. 실험은 700M Hz Pentium CPU, 128M byte main memory의 PC에서 수행되었다. 본 논문에서는 최근접 부근 탐색 방법이 속도 면에서는 이미 검증이 되었기 때문에 이미지의 PSNR 값에 대한 비교분석을 주로 하기로 한다.

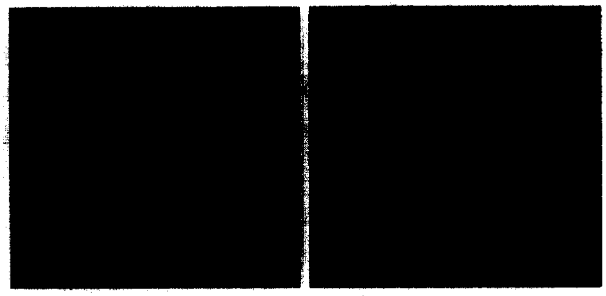


(a) 전체탐색 (b) 제안한 방법

(그림 2) 전체탐색과 제안한 방법의 이미지 비교(Lena)

〈표 1〉 전체탐색과 제안한 방법의 실험결과(Lena)

방법	key의 수	수행시간	PSNR
Full Search		178.23	31.03
제안한 방법	4	7.75	25.07
	8	11.98	27.43
	12	18.13	28.09
	16	24.32	30.90



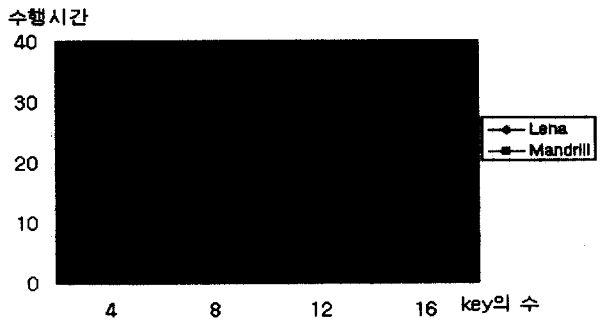
(c) 전체 탐색 (d) 제안한 방법

(그림 3) 전체탐색과 제안한 방법의 이미지 비교(Mandrill)

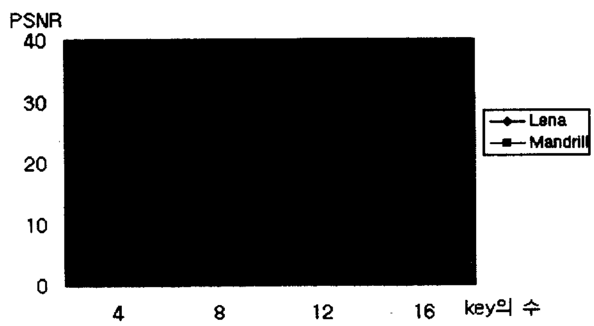
〈표 2〉 전체탐색과 제안한 방법의 실험결과(Mandrill)

방법	키의 수	수행시간	PSNR
Full Search		168.98	28.76
제안한 방법	4	6.31	23.02
	8	10.73	24.92
	12	16.18	26.38
	16	22.58	28.72

〈표 3〉 키의 수와 수행시간 비교



〈표 4〉 키의 수와 PSNR 비교



이상의 결과를 보면 탐색키의 수가 증가됨에 따라 PSNR이 향상되며 수행시간이 늘어난다<표 3>, <표 4>. 실험결과 탐색키가 16개 일 때 Lena의 경우 PSNR이 30.90, Mandrill의 경우 PSNR이 28.72로 각각 전체탐색의 PSNR과 거의 비슷함을 알 수 있고 (그림 2), (그림 3) 과 같이 복원된 이미지 또한 시각적으로 별로 차이가 없음을 알 수 있다. 또한 시스템 환경에 따라 처리속도는 변화될 수 있지만 전처리과정에서 정의역 블록 개수가 선택됨으로써 비교될 블

록 개수가 줄어 압축시간이 단축되었다. 결과적으로 키의 수에 따라 PSNR이 향상되며 키가 16개 일 때 전체탐색과 거의 같은 PSNR을 얻음을 알 수 있다. 여기서, 수행시간은 최근접 부근 탐색 알고리즘의 선택에 따라 더욱 향상 될 수 있다.

5. 결 론

본 논문에서는 정규화 된 분산 값에 대한 최근접 부근 탐색을 하여 유사블록 탐색을 $O(K \cdot \log N)$ 시간에 처리하는 방법을 제안하였다. 본 방법은 부분블록에 대한 독립적인 키 값들을 사용하기 때문에 키의 수를 크게 하면 이미지의 유형에 관계없이 전체탐색과 유사한 PSNR을 가지게 되는 장점을 가진다. 또한, 벡터 변환 방법에 비하여 키 값 계산이 간단하므로 탐색구조를 만들기 전의 전처리 시간이 단축된다.

참 고 문 헌

[1] J. Signes, "Geometrical interpretation of IFS based image coding," Fractals, 5 (Supplementary Issue) : pp.133-143, July, 1997.

[2] Behnam Bani-Equbal, "Enhancing the Speed of Fractal Image," Optical Engineering, Vol.34, No.6, June, 1995.

[3] C. K. Lee, W. K. Lee, "Fast Fractal Image Block Coding Based on Local Variance," 01057-714/98, IEEE, 1998.

[4] S. M. Lee, S. W. Ra, "Fast fractal encoding algorithms using variance of image blocks," Electronics Letters, Vol.34, No. 10, 14th. May. 1998.

[5] R. Hamzaoui, "Codebook Clustering by self organizing maps For Fractal Image compression," Fractals, Vol.2, No.10, 1994.

[6] A. E. Jacquin, "Image Coding Based on a fractal Theory of Iterated Contractive Image Transformation," IEEE Transaction Image Processing, Vol.1, No.1, Jan. 1992.

[7] Gordon Paynter, "Fractal Image Compression," Report on course 0657.420, University of Waikato Oct. 1995.

[8] J. Signes, "Reducing the code-book size in fractal image compression by geometrical analysis," SPIE., Vol.2727, 1996.

[9] J. H. Friedman, J. L. Bentley, R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," ACM Trans. Math Software 3,3, pp.209-226, 1977.

[10] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, A. Wu, "An algorithm for approximated nearest neighbor searching," Proc. 5th Annual ACM Symp. on Discrete Algorithms, pp.573-582, 1994.

김 종 구

e-mail : kjk@mail.jangan.ac.kr

1988년 고려대학교 경영대학원 MIS전공 (경영학석사)

1998년 아주대학교 박사과정 수료

1988년~현재 장안대학 컴퓨터응용계열 부교수

관심분야 : 컴퓨터그래픽스, 데이터베이스

함 도 용

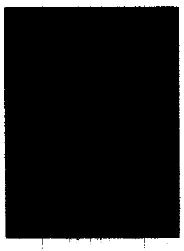
e-mail : detex@korea.com

1980년 중앙대학교 수학과 졸업

1985년 중앙대학교 대학원 수학과 졸업

1996년 아주대학교 박사과정 수료

관심분야 : 컴퓨터그래픽스, 수치해석



위 영 철

e-mail : ycwee@madangl.auou.ac.kr

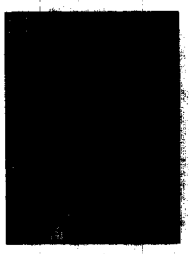
1982년 State University of New York at Albany(B.S.)

1985년 State University of New York at Albany, Computer Science(M.S.)

1989년 State University of New York at Albany, Computer Graphics, Computational Geometry(Ph.D.)

현재 아주대학교 정보통신대학 조교수

관심분야 : Computer Graphics, Computational Geometry Algorithm



김 하 진

e-mail : hjkim@madang.ajou.ac.kr

1962년 서울대학교 문리과 대학 수학과 (이학사)

1978년 Grenoble 1 대학교 대학원 응용수학과 D.E.A.(이학사)

1984년~1985 프랑스 INRIA 초빙교수

1989년~1992년 한국정보과학회부회장 및 회장

1993년~1995년 아주대학교 공과대학장

1974년~현재 아주대학교 정보통신대학 교수

관심분야 : 컴퓨터그래픽스, 수치해석