프랙탈 이미지 압축을 위한 Domain Pool의 분할에 대한 연구

함도용[U]* 김종구* 위영철** 김하진**

아주대학교 대학원 컴퓨터 공학과*
아주대학교 정보 및 컴퓨터 공학부**
dantex@channeli.net

A Study on the Domain Pool Partition for Fractal Image Compression

Do Yong Hahm[U]* Jong-Koo Kim* Young-Choul Wee** Ha-Jine Kimn**

Dept. of Computer Engineering, Ajou University Graduate school*
School of Information and Computer Engineering, Ajou University**

요 약

프랙탈 이미지 압축을 적용하는 경우 많은 시간을 필요로 하는 블록의 비교과정에서 블록 정보의 효율적인 처리를 위하여 선형으로 연결된 기존의 블록 정보 저장을 tree 구조의 형태로 저장하는 방법을 제안한다. Tree의 형태로 저장된 블록의 정보는 BFS 탐색을 이용하여 참조, 사용되며 이를 tree의 각 node에 분류된 domain 블록의 개수로 표현하여 기존의 전체 탐색과 비교 분석하였다.

1. 서론

이미지 정보의 저장과 전송에서 이미지 압축은 중요한 문제이다. Barnsley가 제안하여 이미지 압축에 적용한 IFS(Iterated Function System) 이후 많은 방법들이 이미지의 압축과 저장을 위하여 제안되었다. 자기 유사성의 존재를 가정하여 적용되는 IFS를 이용한 프랙탈 이미지 압축은 축소 사상 원리를 이용하여 원래 이미지의 모든 내용의 저장 대신에 이미지 정보의 변형과 재배치 등에 대한 정보만을 압축 저장하게 된다. 그러나 이때 IFS를 적용하기 위해서 이미지를 재생 복원하는 경우 원래의 이미지를 어느 정도의 품질을 유지하면서 생성하는 것이 가능하도록 할 수 있는 최적의 Affine 변환(Affine Transformation)과 재배치의 방법을 찾는데 많은 수행시간을 필요로 한다. 이미지의 재생에 적절한 변환의 탐색은 이미지를 domain과 range의 영역으로 정의하고 이를 다시 사각형의 형태로 분할(Partition), range R과 비교 되는 domain D에 대한 유사성의 정도를 측정하기 위해 오차 $E = \min_{s,o \in R} \|R-(sD+o)\|^2$를 이용한다. 이러한 오차 E를 최소로 하여 적절한 Affine 변환을 얻는다. 사용되는 오차의 측정은 sup metric에 의한 오차의 측정보다 rms metric에 의한 오차의 측정이 더 효율적이므로 rms metric에 의해 정의되는 오차의 측정 방법을 사용한다[1]. 이미지의 재생에 최적의 Affine 변환 탐색 후 적용과 재배치 후의 이미지는 변환에 사용된 각 블록의 크기에 반비례하여 압축의 속도와 이미지의 품질이 결정된다. A.Jacquin의 PIFS(Partitioned Iterated Function System)와 Fisher의 4등분할 (Quadtree Partition)등의 방법으로 시도된 이러한 자기 유사성 탐색 시간의 단축은 분할된 이미지 블록의 분류를 통해 이루어지는데, 이미지 구성 블록에 대한 특정한 분류 기준에 따른 분류, 블록의 보다 효율적인 저장을 통한 방법과 Polynomial Transformation Term등을 이용하는 분류 방법이 있다[3].

2. Domain의 분류

A.Jacquin의 논문 등에서는 그레이 이미지의 경우 그레이 색깔 값을 각 블록을 비교할 때 유사성 탐색의 효율성과 계산상의 시간 단축을 위하여 거의 같은 형태를 가지도록 고정된 상수 값으로 미리 조정되었다[5].

일반적으로 분할된 이미지 블록은 grayshade 블록과 edge 블록으로 분류되는데, edge의 형태로 분류된 range 블록들만이 domain 블록과의 비교에 사용된다. 그러나 단순한 그레이 이미지의 경우에도 이미지는 자기 유사성(self similarity)이 높지 않아 유사성의 탐색에 많은 시간이 소요된다[1]. 이에 대한 개선으로 Monro와 Dudbridge, 그리고 Oien과 Lepsoy등은 유사성의 비교에 offset의 값을 단순 상수가 아닌 선형 방정식의 형태로 표현, 이미지를 구성하는 색깔의 정보를 재조정하여 적용하는 방법을 제안하였다[6][7].

$o_{ij}=a_1+a_2i+a_3j$로 확장되어 표현되는 o_{ij}는 $o_{ij}=a_1+a_2i+a_3j+a_4i^2+a_5j^2+a_6ij+\ldots$로 일반화되며, scale factor s와 offset o 값을 포함하는 $a_0d_{ij}+\sum_i a_iO_i$ 은 Oien과 Lepsoy등이 처음 사용한 표현으로, Monro와 Dudbridge등은 이를 3차까지 확장 이용하여 이미지 압축에 적용하였다. Range 블록과 domain 블록의 근사 정도를 탐색하고 변환시키는 변환의 탐색 과정에서 domain 블록의 값을 그대로 유지시켜 탐색 과정을 줄이는 시도를 하였다. 본 논문에서는 256x256 그레이 이미지를 사용하였으며 그레이 색깔에 대한 정보를 포함하도록 변환된 데이터는 기하학적인 특성은 계속 포함하게 되어 이미지의 전체적인 특성을 유지하게 되며, 또한 DCT(Discrete Cosine Transform)을 적용, 변환 값을 보다 일반화된 값으로 분류할 수 있게 한다.

이 외의 다른 시도로서 이미지를 마름모의 형태로 블록 분할하여 유사성 탐색시의 오차를 줄이는 시도가 있다. 대각선 방향으로의 이미지 정보 변환이 수평 방향의 이미지 정보보다 시각적으로 더 민감하다는 이론에 근거한 방법으로 H.Deng, N. Xie이 제안하였고 오차의 감소에 어느 정도 개선된 효과를 증명하였다[3].

3. 제안한 방법

많은 수행시간을 요구하는 블록의 탐색 시간을 감소시키기 위하여 range 블록과 비교되는 domain 블록 정보의 전처리를 한다.

[정의] Orthogonal
임의의 두 직선이 만나면 그 교점을 중심으로 4개의 각이 생기며 이 때 한 개의 각이 직각이면 다른 3개의 각 또한 직각이 되는데 이때의 두 직선을 orthogonal (직교)라 한다.

[정의] Orthogonal Projection
각도 θ를 가지고 생성되는 임의의 공간에 수직으로 project 했을 때의 값을 orthogonal project라 한다

D'와 R'를 선형 공간 내에서 정의된, domain 블록과 range 블록의 Orthogonal Projection 결과 값이라 하면 오차 E는 projection 변환 과정을 거쳐 $E=|s_0D'+O'-R'|^2$ 이 된다. D'와 R'는 O'에 대하여 orthogonal이 되며 오차 E는 $|R'|^2-D'\circ R'/|D'|^2$을 이용하여 최소 값을 계산하게 된다[2].

Orthogonality를 이용하여 변환된 식을 블록간의 유사도 측정과 재분류되는 비교 계산과정에 적용하게 되면 실제 계산에서는 contrast와 brightness등이 포함된 복잡한 계산 과정을 배제하는 결과가 되어 계산상의 시간적인 효율을 높일 수 있다. 각 블록의 비교에는 error bound를 분류 기준으로 하여 tree 구조를 적용하였다.

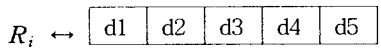
Tree의 각 node는 error bound를 기준으로 보다 작은 오차를 가지는 각 부분 블록의 pixel값에 따라 분류 정렬되며, error bound의 범위를 벗어나는 경우 다음 레벨에서는 정렬된 pixel의 다음 pixel 값들을 기준으로 정렬된다.

Tree의 각 node는 pixel 값을 root로부터 자신이 속한 레벨에 이르는 path에 대한 값으로 가지는 각 부분 블록을 의미한다. 저장된 tree 정보의 탐색은 Breadth First Search를 이용하며 node를 구성하는 각 부분 블록은 주어진 오차에 따라 분류되어 있으므로 한 node가 선택되면 계속해서 다른 레벨의 연결된 node의 부분 블록을 탐색하게 된다.

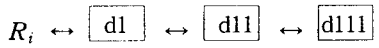
주어진 기준을 충분히 만족하는 branch path는 가장 적합한 node들을 가지는 정보로서 저장되어 압축에 필

요한 정보로 사용된다.

위와 같이 tree 형태로 저장된 정보는 linearly linked 형태의 정보 저장보다는 더 효과적이라 할 수 있다. tree의 각 node로 분류된 domain의 정보는 range의 정보와 비교되어 압축에 사용된다. domain의 정보는 기존의 방법대로 먼저 밝기를 이용한 24가지로 분류되고 이는 다시 variance를 이용한 16가지로 분류 저장된다. 압축 기법의 적용을 위한 탐색 경우 BFS로 탐색되어 비교에 사용, 압축에 사용된다.



< 기존의 방법 >



< 제안한 방법 >

< domain 블록의 저장 형태 >

제안한 방법의 알고리즘은 다음과 같다.

```
Partition the image
    For every domain block
        Apply transformation and quantize
        Insert them into tree
    For every range block
        Set error bound
        Search tree
            Make list of domain blocks
            Until list is empty
        Search the list to have a minimum distance
                between range and domain
```

4.. 검증

Lenna 256x256 8bit/pixel 그레이 이미지를 사용하였다. Range 블록의 크기는 4x4로 하였으며 블록 분할 방법은 Fisher의 Quadtree를 적용하였다. 사용한 시스템의 사양은 셀러론 333A 에 64MB이다.

아래의 표는 한 개의 node를 생성하는 range의 pixel 값에 대하여 대응 참조되는 domain 블록의 개수를 보여주고 있다.

1st class	2nd class	domain block number
0	0-23	155
2	0-23	116
기존의 방법		125000

< 표 1 참조되는 domain의 개수>

5. 결론

본 논문에서는 프랙탈 이미지 압축을 위한 전처리로서, 이미지의 정보를 가지고 있는 블록의 참조와 비교시의 시간 단축을 위하여 기존의 블록 정보 저장을 tree의 형태로 개선하였다. 감소된 블록의 범위는 비교에 사용된 횟수를 비교하였다. 비교횟수의 비교로 시간의 단축을 기대할 수 있었다.. 앞으로의 연구는 블록의 최적화된 크기와 tree의 넓이 및 깊이에 대한 최적화가 될 것이다.

6. 참고문헌

[1] 함도용, 김하진, " PIFS를 이용한 프랙탈 이미지 압축", 컴퓨터 그래픽스 학회 논문집 1997.2

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

[3] H.Deng, N. Xie, W. weng , "A Fast Rhomb Partitioning Fractal Image Compression method", 0-7803-4778-1, IEEE, 1998

[4] R. Hamzaoui, "Codebook Clustering by self - Organizing maps For Fractal Image compression ", Fractals, vol.2,no.0,1994

[5] 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

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

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