

공간영역에서 Quad-tree 분할법을 이용한 적응 화상부호화

Adaptive Image Coding in Spatial Domain Using Quad-tree Segmentation

김태효

경남대학교 전자공학과

Tae Hyo Kim

Department of Electronic Engineering,

Kyungnam University

요 약

본 논문은, 공간영역에서 화상을 압축할 수 있는 Quad-tree 부호화법을 분석하고, 보다 화질 및 압축율을 개선하기 위하여 적응 블록분할 및 병합 알고리즘을 제안하였다. 화상은 에지부분을 제외하고는 인접한 화소들 간에 데이터의 용장도가 높으므로 이 영역을 하나의 대표값으로 설정하여 그 값과 그 블록의 위치좌표를 부호화할 수 있다. Quad-tree 분할은 초기의 병합을 제외하고 순차적으로 분할과정만 반복처리하지만, 본 알고리즘에서는 단위블록(3×3화소)의 평균잔류에너지(MRE)를 이용하여 블록의 분할과 병합을 반복처리한다.

시뮬레이션 결과, 본 알고리즘은 압축율 1bit/pixel에서 기존의 Quad-tree 방법보다 PSNR에서 1.0 dB의 개선이 있었으며, 화상의 블록화 현상도 전혀 나타나지 않았다.

ABSTRACT

In this paper, the quad-tree image compression method being processed in spatial domain is analyzed. In order to improve the quad-tree segmentation, a new method which is used the adaptive block splitting and merging of the quad-tree is proposed. Since image generally has high redundancy between the adjacent pixels except the edge part, this region can be represented to the same value. The values and their data are, hence, available to be encoded and transmitted to the receiving site. The quad-tree segmentation is iteratively processed the splitting of blocks except the initial merging of them, so it is discarded the block which is possible to merge after initial merging. The proposed algorithm is not only applied the iterative splitting, but also the merging after the initial merging using the mean residual energy(MRE) of the unit blocks.

In the simulation, the result showed that the reconstructed image quality was improved as about 1.0 dB under the 1bit/pixel compared with the conventional quad-tree method.

1. 서 론

화상화의 또는 화상전송과 같은 비교적 단순하고 변형성이 적은 화상전송에서, 최근에 많은 관심을 모으고 있는 여러 가지 화상압축 알고리즘이 발표되고 있다[1][2]. 화상데이터는 음성데이터와 같은 일차원적인 데이터구조에 비해 대체로 많은 데이터량을 가지므로 데이터의 저장공간을 크게 요구하며, 실시간으로 화상데이터를 전송하기 위하여 상당한 데이터의 압축이 필요하다.

최근의 화상부호화방식으로 정지화상의 JPEG 및 혼성의 H.261 등은, 변환영역에서 블록단위로 부호화하기 때문에 이들은 블록내의 상관도에 의하여 압축이 이루어지지만, 블록간의 상관도에는 무관하게 되어 블록화현상이 두드러지게 나타나는 단점[3]이 있으며, 또한 변환영역으로 변환하는 시간이 소요된다. 이에 비하여 공간영역에서의 부호화는, 인간의 시각과 일치하는 화상데이터의 값이 유사한 영역단위로 부호화하기 때문에 블록화현상이 없으며, 변환 및 역변환 처리가 필요하지 않으므로 데이터의 처리시간이 단축되는 효과가 있다[4][5].

이와 같이 영역기반의 화상부호화 알고리즘들은 지금까지 여러 가지 고안되었고, 그 중에서 Gargantini [7]에 의해 고안된 Quad-tree 부호화방식은 2진화상을 이용하였는데, 이 방식은 화상의 압축율이 낮았으며 그 후에 Hunter[4] 및 Samet[5] 등에 의해 압축율이 개선되었다. 초기에는 2진화상에 주로 이용되던 이 방식은 중첩영역의 개념을 도입함으로써 능동화상에까지 적용범위가 확장되었다. 능동화상의 Quad-tree방식은 초기 1회에 한하여 분할한 블록에서 동일영역으로 판정되는 블록들을 병합처리하므로 이 후에 분할하는 블록들간에 병합될 수 있는 영역이 생겨도 알고리즘의 구조 때문에 이 부분을 병합할 수 없다 [8][9].

따라서 본 논문에서는, 변환부호화에서 야기되는 블록화현상이 없고 또, Quad-tree방식의 블록병합처리 기능을 개선한 적응 블록병합 알고리즘을 제안하고, 그 특성을 확인하기 위해 표준 시험화상에 적용하여 압축율, PSNR을 기존의 Quad-tree 부호화방식과 비교하였다.

II. Quad-tree 부호화방식

화상의 데이터구조는 변화가 평탄한 영역이 많이 존재하므로, Quad-tree 분할방법은 전체 화상을 나무뿌리와 같은 구조로 분할하는 것으로서 변화가 적은 영역을 단일 블록으로 부호화하고 변화가 큰 영역을 다시 동일 크기의 4개 부분블록으로 분할하여 부호화하는 방식이다. 이방식은 초기에 전체의 화상을 16개의 블록으로 나눈 후, 동일 부모(parent)를 가지는 블록들 중에서 평탄한 영역으로서 병합이 가능한 블록이 있는 경우 이들을 병합처리한다. 다음으로 병합되지 않은 블록들을 대상으로 평탄영역이 아니면 이를 다시 4개의 자식블록들(children block)로 분할하는데, 이 과정을 최소의 단위블록까지 반복한다. 여기서 자식블록으로 분할하는 기준은 병합되지 않는 블록들의 평균잔류에너지(mean residual energy, MRE)를 구하여 주어진 문턱값(threshold value) τ 와 비교한다. 현블록의 MRE가 τ 보다 클 경우 4개의 자식블록들로 분할한다. 이 과정을 그림1에 보였다.

문턱값의 선택은 전체화상의 MRE를 구한 다음에 그 MRE의 비율에 따라 설정하며, 이렇게 결정된 문턱값을 각 블록의 MRE와 비교하므로써 그 블록을 병합 또는 분할할 것인지를 결정한다. 이때 이 문턱값은 복원화상의 화질을 결정하게 된다. 이의 상세한 내용은 문헌 [4]와 같다.

III. 적응 화상부호화

1. 적응 블록분할 알고리즘

기존의 Quad-tree 방식으로 분할하는 경우에 4개로 분할되는 블록들 간에 데이터의 유사성이 존재한다고 해도 이들을 다시 병합처리하지 않고 다음의 분할과정을 계속한다. Quad-tree 분할에서는, 그림2(a)의 A와 B블록과 같이 병합이 가능한 블록이 있어도 부분적으로 병합이 될 수 없지만, 본 논문에서 제안하는 적응 블록분할 알고리즘은 그림(b)의 Q, R 및 S부분과 같이 부분적인 병합이 때 분할과정 후에 이루어진다.

Quad-tree 방식에서는 블록의 병합시에 블록의 MRE를 다시 구하여 병합여부를 판단하지만, MRE를 계산하는 시간이 상당히 소요되므로, 본 알고리즘은 단위블록 MRE를 이용한다. 단위블록 MRE는, 그림3과 같이 블록들을 2×2 (중첩영역을 포함하는 경우 3×3)화소로 잘라서 각각의 잔류에너지(RE)를 구하고, 단위블록의 코너값을 제외한 화소수로 나눈 값이다. 단위블록의 잔류값 $z(i, j)$ 는 식 (1)로 표현할 수 있으며,

$$z(i, j) = R(i, j) - k(i, j) \quad (i, j = 0, 1, 2) \quad (1)$$

여기서, 예측값 $k(i, j)$ 는 식 (2)와 같이 얻어진다.

$$k(i, j) = \frac{(2-i)(2-j)R(0,0) + (2-i) \cdot j \cdot R(0,2) + (2-j) \cdot i \cdot R(2,0) + i \cdot j \cdot R(2,2)}{4} \quad (2)$$

각각의 단위블록 RE $w(i, j)$ 는 식 (3)으로 표현될 수 있고,

$$w(i, j) = \frac{\sum z^2(i, j)}{5} \quad (3)$$

단위블록 MRE는 식(4)와 같이 계산된다.

$$E[w(i, j)] = \frac{\sum_{i=0}^2 \sum_{j=0}^2 w_n(i, j)}{n} \quad (4)$$

적응 블록분할 및 병합과정은 그림3과 같이 현재 블록을 4개의 자식블록들로 분할하고 분할된 4개 블록들의 MRE를 구하여 주어진 문턱값 τ 와 비교한다. 이때 $MRE \geq \tau$ 이면 다시 4개의 손자블록들로 분할하며, 이들 중 유사한 블록이 있으면 이들을 병합한다. 만일 $MRE < \tau$ 이면 이 블록은 분할과정이 종료되어 다음 블록의 분할과정으로 연결된다. 이러한 과정을 단위블록이 될 때까지 반복한다. 이 과정에서 병합된 영역의 블록에 있는 대부분의 데이터는 제거되며, 단지 그 블록의 대표값만이 남게된다. 그림 4(a)와 같이 중첩블록(overlapped block)의 흑색부분은 대표값으로 남게되는 부분이고, 나머지 백색부분의 영역은 사라지게 된다. 블록의 좌측 상부 코너의 흑색부분이 대표값과 그 블록의 위치데이터가 수신측으로 전송되며 수신측에서는 전송된 블록들의 위치데이터와 대표값으로부터 제거된 화상데이터를 복원하게 된다.

2. 화상복원 알고리즘

전송된 블록들의 위치데이터와 해당 블록의 코너에 있는 대표값들로부터 화상을 복원하게된다. 먼저 전송된 화상블록들의 데이터가 그림4(a)와 같고 하면, 이로부터 보간법을 이용하여 가운데 흰부분의 값들을 단계적으로 예측해 가며 이 과정은 그림(b), (c), (d)와 같이 이루어진다. 복원과정에서 쌍일차보간(bilinear interpolation) 예측값은 코너의 대표값을 이용한 선형 보간법에 의해 복원된다.

선형 보간법은 $N(N=2^n+1)$ 개의 샘플에 대하여 식 (5)와 같이 수행된다.

$$x(k) = \frac{(N-k)x(0) + kx(N)}{N} \quad k=0, \dots, N \quad (5)$$

여기서, 중심값은 식(6)와 같다.

$$x\left(\frac{N}{2}\right) = \frac{x(0) + x(N)}{2} \quad (6)$$

그리고 식 (4)를 다시 반주기로 분리하면 각각 식 (7)과 (8)로 표현할 수 있다.

$$x(k) = \frac{\left(\frac{N}{2} - k\right)x(0) + kx\left(\frac{N}{2}\right)}{\frac{N}{2}} \quad k=0, \dots, N/2 \quad (7)$$

$$x\left(\frac{N}{2} + k\right) = \frac{\left(\frac{N}{2} - k\right)x\left(\frac{N}{2}\right) + kx(N)}{\frac{N}{2}} \quad k=0, \dots, N/2 \quad (8)$$

그림5에서 알 수있는 것과 같이 식 (8)에 의하여 블록의 모든 화소값이 구해질 수 있도록 보다 작은 블록에 이르기까지 계산한다.

이와 같은 원리에 의하여 화상복원의 쌍일차 보간 방정식은 식 (9)과 같다.

$$a(i, j) = \frac{A(N-i)R(N-j) + B(N-i) + C + i(A(N-i) + B) + j(C(N-j) + D)}{N} \quad (9)$$

이를 다시 재정리하면 식 (10)와 같다.

$$a(i, j) = \frac{(N-i) \left\{ \frac{A(N-j) + B}{N} \right\} + j \left\{ \frac{C(N-j) + D}{N} \right\}}{N} = \frac{(N-i)R(N-j) + B(N-i) + C + j \left\{ \frac{C(N-j) + D}{N} \right\}}{N} \quad (10)$$

식(10)의 $x(i, j)$ 값은 네개의 코너에 있는 대표값으로부터 얻을 수 있다.

IV. 시뮬레이션 및 결과

본 논문에서 제안한 적응 화상부호화 알고리즘의 유용성을 시험하기 위하여 다음과 같은 과정에 따라서 처리하였다.

- (1) 시뮬레이션 프로그램을 그림5에 따라 구성하여, 8비트 데이터의 256×256 표준 시험화상(Lenna)을 모의실험 하였다.
- (2) 각각의 블록에서 단위블록 MRE를 구하여 설정된 비트율에 맞도록 문턱값 τ 를 설정하였다.
- (3) 문턱값 τ 를 기준으로 하여 블록을 분할 및 병합하고, 각 블록의 대표값과, 분할 및 병합과정에서 해당블록들의 위치데이터를 분할 목록으로 작성하여 부호화하였다.
- (4) 8비트 농담값(gray level)을 갖는 대표값을 5비트의 농담값으로 변환시켰다. 이때 비트율은 식(11)과 같이 계산할 수 있다.

$$rate_{bpp} = \frac{sam + q \cdot 5}{256 \cdot 256} \quad (11)$$

여기서 sam은 분할목록의 수이고 q는 대표값이다.

- (5) 대표값과 분할목록의 정보를 이용하여 각 블록의 해당위치에 대표값을 위치시켰다.
- (6) 분할목록을 이용하여 쌍일차 보간법으로 화상을 복원하였다.
- (7) 모의실험 결과는 식(12)로부터 PSNR을 계산하였다.

$$PSNR = 10 \cdot \log_{10} \left[\frac{255^2}{\sum_{i,j} (X_i - X'_i)^2 / M} \right] \quad (12)$$

여기서 X_i, X'_i 는 각각 원화상의 화소와 복원화상의 화소값을 나타내고 M 은 부호화된 화상내에 있는 화소의 수이다.

이와 같은 과정의 시뮬레이션을 통하여 얻어진 원화상에 대한 복원화상의 PSNR특성을 기존의 Quad-tree 부호화 알고리즘의 결과와 비교하여 표(1)에 나타내었다.

표1. Lenna 화상의 시험결과

Bit rate	Threshold value : τ	PSNR dB
1 bpp	174 (207)	30.1 (29.0)
0.5bpp	529 (691)	26.8 (26.04)

시뮬레이션 결과에서, 화상의 압축율을 각각 1bit/pixel 및 0.5bit/pixel로 하였을 때의 화질평가를 본 알고리즘과 기존의 Quad-tree법과 비교해 볼 때, 제안된 본 적응 알고리즘은 기존의 Quad-tree 방법보다 PSNR이 약 1dB정도 개선 되었음을 알 수 있었다. 본 알고리즘에서 얻은 화질은 주파수영역에 처리기반을 둔 ICT의 경우와 비슷한 결과이지만, DCT에서는 공간영역에서 주파수영역으로의 변환과 역변환이 필요하므로 많은 계산시간을 요하고, 또 화상의 블록화현상을 피할 수 없다. 이에

비하여 본 알고리즘에서는 블록화 현상은 전혀 나타나지 않는다.

그림6에 각각 기존의 Quad-tree법과 제안된 알고리즘에 의해 복원된 영상을 나타내었다.



(a) Original image 8bpp



(b) Quad-tree : $\tau=189, 1.0\text{bpp}$



(c) 제안된 방법 : $\tau=176, 1.0\text{bpp}$



(d) Quad-tree : $\tau=691, 0.5\text{bpp}$



(e) 제안된 방법 : $\tau=529, 0.5\text{bpp}$

그림6. 화상복원 결과



(a) Quad-tree방식의 대표값(0.5bpp)



(b)제한된 방식의 대표값(0.5bpp)

그림7. 전송된 화상의 대표값

그림7은 나누어진 블록에서 각 블록의 대표값만이 남고 나머지는 제거된 영상이다. 여기서 그림7의 (a)보다 (b)가 경계부분(변화성분이 많은 부분)에 더욱 많은 점들이 모여 있기 때문에 동일한 압축률에서 제한된 방식의 화질이 더욱 우수함을 알 수 있었다.

V. 결론

본 논문에서는 공간영역에서 영상을 압축 하는 최우 블록 분할 및 병합 알고리즘을 제안하였다. 종래의 Quad-tree에서는 순차적인 블록의 분할 과정만 반복 처리하였으나, 이 과정에서 병합되어질 수 있는 영역을 무시하게 된다. 이에 비하여 제안된 알고리즘은 분할과 병합을 매 과정마다 병행하며, 단위블록의 잔류에너지를 이용하였다. 본 알고리즘의 특성을 평가하기 위해서 기존의 Quad-tree 방식과 비교하였다.

모의실험 결과, 동일한 압축율 1bit/pixel에서 기존의 Quad-tree 방식에서는 29dB, 제안된 방법에서는 30dB로 나타나 1dB정도 개선되었다. 이 결과는 표준 DCT의 PSNR과 비슷한 수준이지만, DCT 보다 처리 시간이 빠르고 블록화 현상이 나타나지 않는 우수한 특성을 가지고 있음을 알 수 있었다.

참고문헌

1. A. K. Jain, "Image Data Compression A Review," Proc. IEEE, vol.69, no.3, pp. 349-389, Mar. 1981.
2. Pavlidis, T., Algorithms for Graphics and Image Processing, Computer Science Press, Rockville, MD, 1982.
3. T. Saito, et al, "Adaptive Discrete Cosine Transform Image Coding Using Gain/Shape Vector Quantizers," pp. 129-132, ICASSP 86.
4. G. M. Hunter, and K. Steiglitz, "Operation on Images Using Quadtree Nodes," IEEE Trans. Pattern Anal. mach. Intell. Vol. PAMI-1, No.2, pp. 145-153, April 1979.
5. Samet, H., The Design and Analysis of

Spatial Data Structures, Addison-Wesley Publishing Company, Inc., MA., U.S.A., 1990.

6. C. A. Shaffer, and H. Samet, "Optimal Quadtree Construction Algorithm," Computer Vision, Graphics and Image Processing, Vol. 37, No.3, pp.402-419, March 1987.
7. I. Gargantini, "An Effective Way To Represent Quadtrees," Communications of ACM, vol. 25,N.12, pp. 905-910, December 1982.
8. Farrelle, P. M., Recursive Block Coding for Image Data Compression, Springer-Verlag, 1990.
9. Kernighan, B. W. and Ritchie, D. M., The C Programming Language, Prentice-Hall, EngleWood Cliffs, New Jersey, 1978.
10. Farrelle, P. M and Jain, A. K., "Quad-tree Based Two Source Image Coding," MILCOM86,49.2.1-49.2.5, Monterey, CA, Oct. 1986.
11. M. N. Ganbegan, "An efficient use of quadtree in geographic information systems," International Journal of Geographic Information Systems, Vol. 3, pp. 201- 214, 1989.
12. W. K. Pratt, Image Transmission Techniques, New York: Academic, 1979.

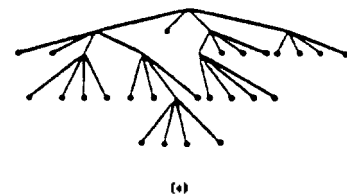
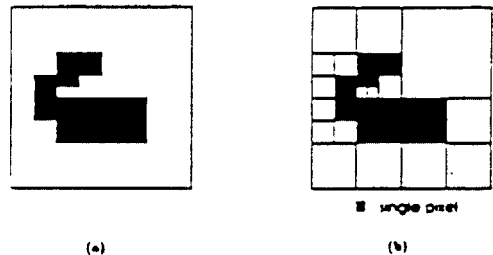


그림1. Quad-tree 분할의 원리

A	B	I		
C0 C1	D0 D1			
C2 C3	D2 D3			
I0 I1	J	M	L	
I2 I3	K	L	O	P

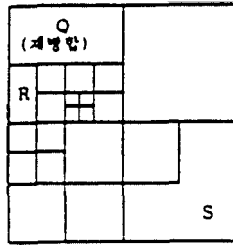


그림2. Quad-tree 및 좌우 분할법의 예

12	18	11	21	20
10	9	7	11	7
7	17	32	9	9
11	24	5	8	11
18	9	12	8	9

□ 단위블록의 모서리값

(a)

12	18	11	11	21	20	7	17	32	32	9	9
10	9	7	7	11	7	11	24	5	5	8	11
7	17	32	2	9	9	18	9	12	12	8	9

(b)

12	11	11	11	15	20	7	19	32	32	30	9
9	15	21	21	17	14	12	12	23	23	15	9
7	19	32	32	30	9	18	14	12	12	10	9

(c)

0	2	0	0	6	0	0	-2	0	0	-11	0
1	-4	-14	-14	-6	-7	-1	12	-17	-17	-7	2
0	-2	0	0	-11	0	0	-5	0	0	-2	0

(d)

그림3. 단위블록의 MRE 산출방법

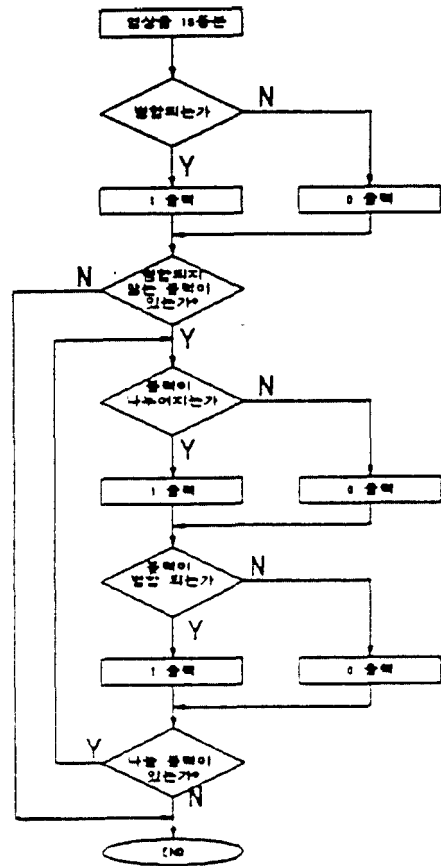
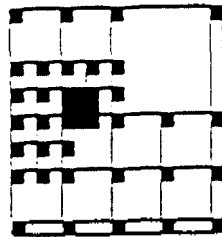
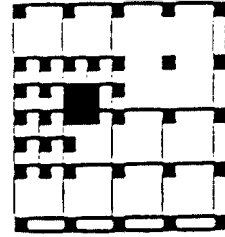


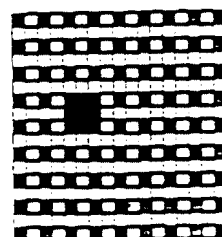
그림5. 좌우 알고리즘의 흐름도



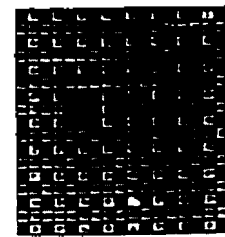
(a)



(b)



(c)



(d)

그림4. 보간을 이용한 화상복원의 원리