

Breadth First Gray Quadtree : 화상의 효율적 표현법

BF Gray Quadtree : Efficient Image Representation Method

李 克* · 李 玟 奎** · 黃 熙 隆*** · 李 廷 元§
 (Geuk Lee · Min-Gyu Lee · Hee-Yeung Hwang · Jung-Won Lee)

요 약

본 논문에서는 이진화상을 4진 나무 구조로 표현하는 새로운 방법(BFQT)를 제안한다. BFQT는 나무 구조를 넓이 우선으로 탐색하여 혼합 노드(gray node)의 자식 노드 패턴을 십진수로 표현한다. 실제 화상을 이용해 다른 표현법들과 비교 실험하여 그 타당성을 보인다.

Abstract- A new compact hierarchical representation method of binary image is proposed. This method represents a binary image with the set of decimal numbers. Each decimal number represents the pattern of nonterminal node(gray node) in the quadtree. This pattern implies the combination of its four child nodes. The total number of gray nodes is one third of that terminal nodes. We show that gray tree method is efficient comparing with others which have been studied.

1. 서 론

이진 화상의 효율적인 표현방법은 화상처리(image processing), 컴퓨터 비전(computer vision), 지도분석(geographic analysis) 등의 분야에 매우 중요한 부분이다. [2] 즉 화상의 코딩방

법을 개선함으로써 저장 정보량의 감축, 통신정보량의 감소와 여러가지 화상 연산을 쉽게하여 주는 효과가 있으며, 더구나 이진 화상의 표현방법은 3차원 물체 표현과 다치영상의 표현에 기본이 되는 코딩방식으로 그 파급효과가 크다.

초기 이진 영상표현은 비트 맵 방식이나 run-length 코드 방식이 이용되었으나 다양한 화상의 처리 필요성이 증가됨에 따라 화상을 구조적이고 계층적으로 표현하는 방법이 연구되었으며 계층적 표현법은 기억장소를 축소시키고 수행시간을 개선시킨다. [6, 9, 11]

계층적인 화상자료 표현법의 대표적인 예가 4진 나무 구조(Quadtree) 표현법으로서 그동안 여러

*正 會 員 : 韓南大 工大 電子計算工學科 助教授
 **正 會 員 : 仁聖시스템 理事
 ***正 會 員 : 서울대 工大 컴퓨터工學科 教授 · 工博
 §正 會 員 : 弘益工業專門大 電子計算學科 · 副教授
 接受日字 : 1989年 6月17日
 1次修正 : 1990年 3月24日

사람들에 의해서 연구되어 왔다. [1, 3, 5, 7, 11, 12] 4진 나무 구조는 화상을 연속적으로 4등분하여 최소의 단위인 화소에 이를 때까지 분할한 뒤 동일정보 (1: 검은색 또는 0: 흰색)의 4의 지수승 갯수로 이루어진 블록들과 화소들로 표현하는 방법이다. 뿌리노드(root node)는 전체 화상을 나타내며 각 노드의 자식 노드(child node)는 부모노드(parent node)의 1/4 크기의 화상을 나타낸다. 혼합노드(nonterminal node: gray node)는 4개의 자식 노드를 가지고 있다. 각 단말노드(terminal node)는 같은 정보 ('1' 또는 '0')를 갖는 전체 화상의 부분 집합을 나타낸다.

4진 나무 구조의 방향성과 경로에 의한 주소 지정 방식의 예는 Fig.1과 같으며 특정화상을 4진 나무 구조로 표현한 예는 Fig.2와 같다. 여기서 G는 중간노드를 나타낸다.

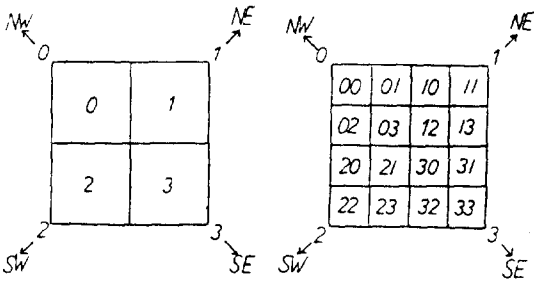


그림 1 4진 나무의 방향 및 주소 할당
Fig. 1 Direction and Address Allocation of Quadtree

2. 4진 나무 구조의 여러가지 표현법

2.1 정규 4진 나무 구조(Regular Quadtree)

정규 4진 나무 구조 표현법은 검은노드, 흰노드, 중간노드를 6개의 영역으로 표현하는 방법으로 1개의 부모노드를 가르키는 포인터와 4개의 자식노드를 가르키는 포인터, 자신의 노드 종류(흰노드, 검은노드, 중간노드)를 나타내는 필드를 갖는 레코드로 구성된다. 정규 4진 나무 구조의 코딩 방법과 예는 Fig.3과 같다. 하나의 노드가 가지는 6개 필드의 의미는 다음과 같다.

NK: 노드의 종류를 나타내는 필드

PP: 부모노드를 나타내는 포인터

S₀P, S₁P, S₂P, S₃P: 4개의 자식노드를 나타내는 포인터

이 방법은 모든 노드를 모두 표현해야만 되고 여러개의 포인터가 필요하기 때문에 많은 양의 기

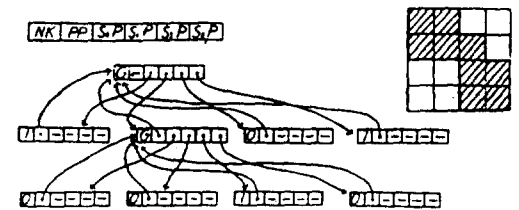
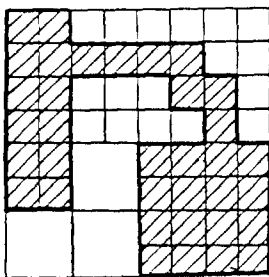
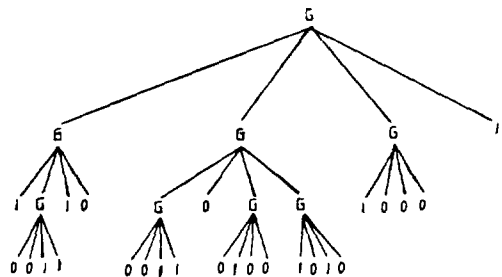


그림 3 그림 3 정규 4진 나무 구조 표현법
Fig. 3 Regular Quadtree Representation



(a)



(b)

그림 2 (a) 이진 화상 I
(b) 그림 2(a)의 4진 나무 표현

Fig. 2 (a) Binary Image I
(b) Quadtree Representation of Fig.2(a)

억 장소가 필요하다.

2.2 선형 4진 나무 구조(Liner Quadtree)

선형 4진 나무구조는 포인터로 인한 기억장소의 낭비가 많아지므로 검은 노드만을 저장하고 포인터없이 암시적으로 해당 노드에서 뿌리 노드까지의 경로를 내포하는 코딩 방식이 제안되었으며 이 결과 화상 자료를 검은 노드의 수에 비례하는 시간과 공간 복잡도 알고리즘으로 표현할 수 있게 되었다. [7]

Fig.2(a)의 화상을 선형 4진 나무 구조를 표현한다면 다음과 같이 된다.

3XX 00X 02X 20X 012 013 102 103 121 130 132

선형 4진 나무 구조 표현법은 특수 문자 'X'를 사용하며 이 때문에 기억장소를 효율적으로 사용하지 못하게 된다.

2.3 깊이 우선 4진 나무 구조(Depth First Quadtree)

이 방법은 문맥자유문법(context free grammar)를 이용해 좌측 부속 나무 구조(left subtree)로 부터 우선적으로 코딩하므로 깊이 우선 표현법이라 한다. 이 방식의 코딩에서는 나무 구조의 깊이를 나타내기 위해 '('라는 문자를 사용하며 검은 노드를 나타내기 위해 '1', 흰 노드를 나타내기 위해 '0'을 사용한다. [5]

Fig.2(a) 화상을 깊이 우선 표현법으로 표시하면 아래와 같이 된다.

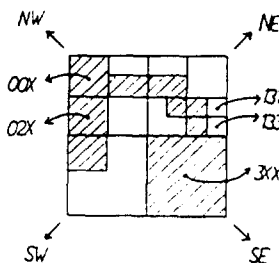


그림 4 선형 사진 나무 구조의 주소 할당
Fig. 4 Address Allocation of Linear Quadtree

((1 (001110 ((00110 (0100 (1010 (10001
이것은 fig.2(b)의 트리에서 G를 '('로 바꾸어 깊이가 우선으로 탐색한 결과와 동일하다.

2.4 넓이 우선 4진 나무 구조(Breadth First Linear Quadtree)

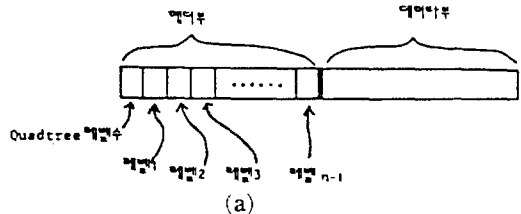
선형 4진 나무 구조 표현법에서 특수 문자인 'X'를 제거함으로써 화상 정보의 양을 대폭 감소시킨 코딩 방식이다. [1]

이 방식은 4진 나무구조의 각 레벨의 노드 수에 대한 정보를 가지고 있는 고정된 크기의 헤더부(header)와 노드의 형태에 대한 정보를 가진 데이터부로 구성된다. 이 코딩 방식은 검은 노드만 표현하며 헤더부의 정보에 의해 포인터 없이 각 노드에서 뿌리 노드 까지의 경로를 알 수 있다.

자료 구조와 코딩의 예를 Fig.5에 보인다.

3. 넓이 우선 혼합 4진 나무 구조(Breadth First Gray Quadtree)

혼합노드(gray node)란 검은노드를 제외한 중간노드(nonterminal node)를 의미한다. 임의의 화상을 4진 나무 구조로 표현했을 경우 혼합노드의 개수는 단말 노드의 개수의 1/3에 지나지 않는다. 깊이 우선 표현법은 4진 나무 구조로 표현된 화상을 깊이 우선 탐색하여 탐색 순서대로 모든노드(검은노드, 흰노드, 혼합노드)를 '1', '0', '('로 표시하나 넓이 우선 혼합 표현법은 4진 나무 구조를 넓이 우선으로 탐색하여 탐색 순서대로 나타나는 혼합노드가 가지는 자식 노드의 패턴을 10진수로



4 1 3 7 | 3 | 00 02 20 | 012 013 102 103 121 130 132
(b)

그림 5 (a) BF 선형 4진 나무의 자료구조
(b) 그림2(a)의 BF선형 표현법

Fig. 5 (a) Data Structure of BF Linear Quadtree
(b) BF Linear Representation of Fig.2 (a)

표현한다.

트리내의 혼합 노드들은 4개의 자식 노드를 가지며 각각의 자식 노드는 검거나 희거나 혹은 혼합 노드일수 있으므로 81가지 (3⁴가지) 패턴중의 하나이며 이 중 나타나지 않는 2가지 경우(0000과 1111의 경우)를 제외하면 79가지 패턴중의 하나이다. 79가지의 서로 다른 패턴을 표현하기 위해서는 7비트가 필요하며 따라서 하나의 혼합 노드를 표현하기 위해 7비트가 필요하다.

그림 6의 (a) 화상의 트리 표현이 (b)이다(여기서 2는 gray node를 나타낸다). (b)를 깊이 우선으로 표현하면 다음과 같다.

DF : (1 (00100 ((1011010

'0', '1', ') 표현이 각각 2비트씩 필요하다면 34비트가 필요하다. 넓이 우선으로 표현하는 경우 첫번째 혼합 노드인 뿌리 노드(root node: level 1)의 자식 노드 패턴은 '1202'라는 3진수 표현이 된다. 이를 10진수로 바꾸면 47이다. 레벨 2에는 2개의 혼합 노드가 있고 각각 '0010'과 '2010'의 자식 노드를 가진다. 10진수로 바꾸면 03과 57이 된다. 레벨 3은 1개의 혼합 노드가 있고 '1011'의 비트패턴이므로 31이 된다. 레벨 4에는 혼합 노드가 없다. 이들을 나열하면 아래와 같다.

BF : 47 03 57 31

2차리의 10진수 표현에 7비트가 필요하므로 28비트가 소요된다. 화상의 크기(size)를 안다면 (n-1)레벨에 있는 노드들은 '0'과 '1'로만 구성된 노드 패턴을 가지므로 4비트로 표시할 수 있다.

4. 비교 및 평가

4.1 화상 압축에 필요한 기억용량

여러가지 코딩 방식으로 표현된 화상의 기억 용량에 관한 모델 즉 화상 기억에 필요한 총 비트수는 다음과 같다.

M : total number of terminal nodes in quad-tree

B : total number of black nodes in quad-tree

W : total number of white nodes in quad-tree

n : number of level (resolution parameter)

i : level i

G_i : gray node number at i-th level

B_i : black node number at i-th level

4.1.1 Linear Quadtree

(3*(n-1)+2)*B Bits

0, 1, 2, 3의 문자 표시에 2비트가 필요하며 'X'

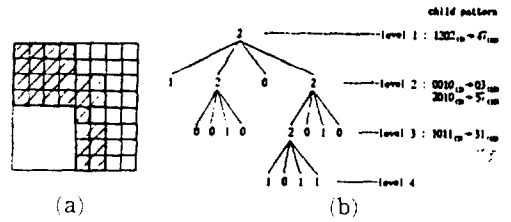


그림 6 (a) 이진 영상 II

(b) 그림6(a)의 4진 나무 표현

Fig. 6 (a) Binary Image II

(b) Quadtree Representation of Fig.6(a)

표현에 3비트가 필요하다. 따라서 하나의 검은 노드를 나타내는 스트링에 3(n-1)+2비트가 필요하다. [7]

4.1.2 Depth First Expression

system 1 : 2*[M+(M-1)/3]Bits

system 2 : {2*[(B+W-1)/3+B]+W}Bits

시스템1의 경우 0, 1, (중 하나가 트리내의 모든 노드의 갯수 만큼 나타나며 그 하나를 표시하는데 2비트가 필요하다. 시스템2의 경우 혼합 노드와 검은 노드는 2비트로 흰 노드는 1비트로 표시된다 ('0'는 11로, '1'은 10으로 '0'은 0으로 표시). [5]

4.1.3 BF Linear Quadtree

[ln(n-1)+∑_{i=1}ⁿ⁻¹2*(n-1)+2*∑_{i=1}ⁿ⁻¹(i*B_i)]Bits

허더부에서 Quadtree 레벨수를 표시하는데 ln(n-1)비트, 각 레벨에서 검은 노드의 갯수를 표시하는데 ∑_{i=1}ⁿ⁻¹2*(n-1)비트, 데이터부에서 0, 1, 2, 3중 하나를 표시하는데 2비트에 레벨에 따라 검은 노드의 수가 많아지므로 ∑_{i=1}ⁿ⁻¹(i*B_i)*2비트가 필요하다. [1]

4.1.4 BF Gray Quadtree

[7*∑_{i=1}ⁿ⁻²(G_i)+4*G_{n-1}]Bits

레벨 1에서 (n-2)레벨 사이의 혼합 노드는 7비트, (n-1)레벨의 혼합 노드는 4비트가 소요된다. 화상의 압축율은 다음과 같이 정의한다. 압축율=

$\frac{\text{원래 화상을 나타내는데 필요한 총 비트수}}{\text{코딩된 화상 정보의 총 비트수}}$

4.2 최악의 경우의 성능 비교

자료 압축율이 가장 나쁜 경우 즉 나무 구조가

포화 4진 나무 구조(full Quadtree)를 이룰 때 총 정보 양의 수렴값을 구해 본다. 이 경우 전체 트리는 4^n 개의 단말노드들로 이루어진다. 원래의 화상 정보는 4^n 비트로 표현되며 $[(4^n-1)/3] \approx 4^n/3$ 개의 혼합 노드와 각각 $4^{n/2}$ 개의 검은 노드와 흰 노드를 갖는 포화 4진 나무구조라고 가정하면 각 코딩 방식에서 다음과 같은 식이 나온다.

4.2.1 Linear Quadtree

$$1/2 \cdot 4^n \cdot (3 \cdot (n-1) + 2) = (3/2) \cdot n \cdot 4^n - 1/2 \cdot 4^n \text{ Bits}$$

예를 들어 $n=8$ 이라고 가정하면 원래 화상의 약 11.5 배의 정보양이 필요하다.

4.2.2 Breadth First Linear Quadtree

$$[\lceil \ln(n-1) \rceil + \sum_{i=1}^n 2^i \cdot (n-1) + 2^i \cdot (n-1) \cdot 4^{n/2} / 2] \text{ Bits}$$

레벨 1에서 부터 레벨 $(n-2)$ 까지에 검은 노드는 존재하지 않으므로 데이터부의 합산항은 $(n-1)$ 레벨에 $4^{n/2}$ 개의 검은 노드만으로 표시된다. $n=8$ 이라 가정하면 원래 화상의 약 7배 이상의 정보가 필요하다.

4.2.3 Depth First Expression

$$\text{system1} : 2 \cdot [4^n + 4^{n/3}] = 2.7 \cdot 4^n \text{ Bits}$$

$$\text{system2} : 2 \cdot [4^{n/3} + 4^{n/2}] + 4^{n/2} = 2.17 \cdot 4^n \text{ Bits}$$

n 에 무관하게 각각 2.7 및 2.17배의 정보가 필요하다.

4.2.4 BF Gray Quadtree

$$n-1 \text{ 레벨의 노드수} : 4^{n-1}$$

$$n-2 \text{ 레벨까지의 총 노드수} : (4^{n-1}-1)/3 \approx 4^{n-1}/3$$

$$4 \cdot 4^{n-1} + (4^{n-1})/3 \cdot 7 = 1.58 \cdot 4^n \text{ Bits}$$

포화 4진 나무 구조이므로 $(n-1)$ 레벨 까지의 모든 노드는 혼합 노드이다. 따라서 이것은 레벨 변수에 무관하게 약 1.58 배의 정보가 필요하다.

4.3 실험을 통한 비교

비교분석을 위해 5개의 화상을 Data Translation사의 DT 2803 시스템을 이용해 256×256 gray image를 받아 이진 화상으로 변환하여 사용하였다. 실험에 이용된 화상은 Fig.7과 같으며 비교한 결과는 Table 1에 있다.

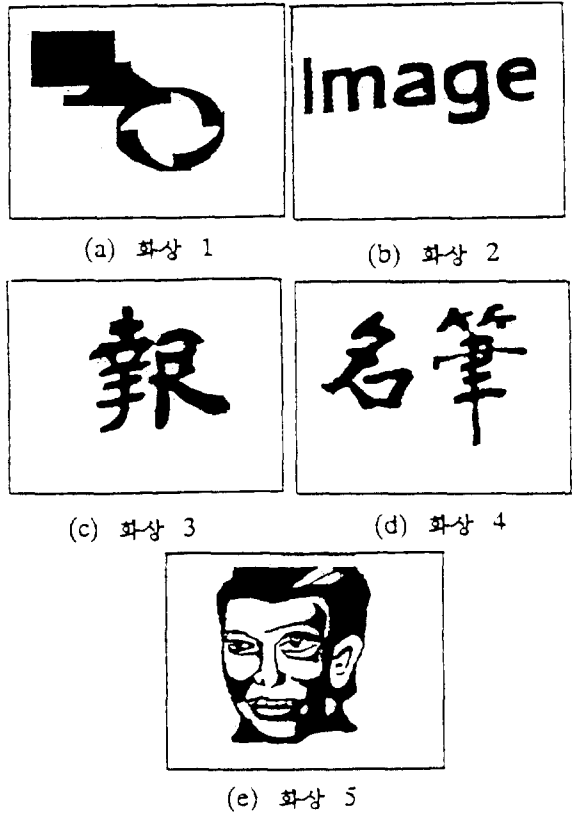


그림 7 표본 영상 (a)~(e)
Fig. 7 Sample Image (a)~(e)

표 1 코딩 방법들의 비교
Table 1 Comparison of Coding Methods

화상의 종류		화상1	화상2	화상3	화상4	화상5
Linear	bits	19067	33235	31395	36340	62284
Quadtree	압축율	3.44	1.97	2.09	1.80	1.05
BF Linear	bits	11929	21579	20233	23373	40353
Quadtree	압축율	5.40	3.04	3.24	2.08	1.62
DF Quadtree (System1)	bits	3880	6636	5876	7001	11779
	압축율	16.89	9.88	11.15	9.36	5.56
BF Gray Quadtree	bits	3652	5967	5090	6265	10253
	압축율	17.95	10.98	12.88	10.46	6.39

5. 결 론

화상을 4진 나무구조로 표현함으로써 화상 정보의 양을 줄일 수 있음을 살펴 보았다. 특히 넓이 우선 혼합 4진 나무 표현법은 실제 화상이나 압축률이 낮은 경우에도 높은 적응성을 가짐을 실험과 최악의 경우식을 비교하여 그 타당성을 보였다. 넓이 우선 4진 나무 구조 표현법을 응용하여 3차원 물체 표현 및 다치 화상의 영역으로 쉽게 확장할 수 있으며 좋은 결과가 예상된다. 본 방법이 여러가지 연산(union, masking, 반전, 면적측정...)에도 쉽게 이용됨을 앞으로 보이려 한다.

참 고 문 헌

- [1] 이극, 한상호, 김민환, 황희용, "BF 선형 4진 트리로 표현된 화상에서 둘레 길이 연산", 대한 전기학회 창립 40주년 기념 학술대회 논문집, pp. 334-337, 1987.
- [2] Azriel Rosenfeld, Avinash C. Kak, Digital Picture Processing, vol. 2, pp. 191-204, 1982.
- [3] Chris L. Jackins, Steven L. Tanimoto, "Quadtree, Octtree and K-trees: A Generalized Approach to Recursive Decomposition of Euclidean Space", IEEE Trans. PAMI, vol. PAMI-5, no. 5, pp. 533-539, Sep. 1983.
- [4] Dana H. Ballard, Christopher M. Brown, Computer Vision, Prentice-Hall, pp. 249-252, 1982.
- [5] Eiji Kawaguchi, Tsutomu Endo, "On a Method of Binary-Picture Representation and Its Application to Data Compression", IEEE Trans. PAMI, vol. PAMI-2, no. 1, pp. 27-35, Jan. 1980.
- [6] Hanan Samet, "Distance Transform for Images Represented by Quadtrees", IEEE Trans. PAMI, vol. PAMI-4, no. 3, pp. 298-303, May. 1982.
- [7] Irene Gargantini, "An Effective Way to Represent Quadtrees", Comm. ACM, vol. 25, no. 12, pp. 905-910, Dec. 1982.
- [8] J.R. Woodwark, "The Explicit Quadtree as a structure for Computer Graphics", The Computer Journal, vol. 25, no. 2, pp. 235-238, 1982.
- [9] M.A. Oliver Wiseman, "Operations on Quadtree Encoded Images", The Computer Journal, vol. 26, no. 1, pp. 83-91, 1983.
- [10] Markku Tamminen, "Encoded Pixel Trees", Computer Vision, Graphics and Image Processing, vol. 28, pp. 44-57, 1984.
- [11] Michael Shneier, "Two Hierarchical Linear Feature Representations: Edge Pyramids and Edge Quadtrees", Computer Graphics and Image Processing, vol. 17, pp. 211-224, 1981.
- [12] Yoav Cohen, Michael S. Landy, M. Pavel, "Hierarchical Coding of Binary Images", IEEE Trans. PAMI, vol. PAMI-7, no. 3, pp. 284-287, May. 1985.