이진 영상을 위한 Compact Complementary Quadtree의 구성

조 영 우† · 김 영 모†

요 약

본 논문에서는 상보 쿼드트리에 기반한 새로운 이진 영상의 preorder tree traversal 코딩 방법인 Compact Complementary Quadtree(CCQ)를 제안한다. 제안한 방법에서는 쿼드트리 내의 마디를 표시하기 위해서 G, B, W의 심볼을 사용하는 대신에 9개의 형태부호(type code)를 이용한다. 실험을 통해서 CCQ가 DF-expression 방법보다 압축 효율이 뛰어남을 확인할 수있다. 제안한 방법은 line drawings, 지리 정보 영상, 그리고 그레이 레벨 영상의 이진화 영상 등 이진 영상의 계층 구조화와 점진적 전송에 효과적으로 이용될 수있다.

Compact Complementary Quadtree for Binary Images

Young-Woo Cho † · Young-Mo Kim †

ABSTRACT

In this paper, we propose a new preorder tree traversal method for binary images, named the Compact Complementary Quadtree (CCQ). In the proposed method we use type codes for representing nodes in the quadtree instead of using the symbols G, B, and W. From the experimental results, we have confirmed that the CCQ has a higher compression ratio than that of the DF-expression. CCQ can be effectively applied to progressive transmission of binary images such as line drawings, geographical maps, and halftones.

1. Introduction

The quadtree[1-4] is a hierarchical data structure based on a regular decomposition of space. The investigation of hierarchical data structure has also been concerned with how to encode the tree. The most simple quadtree encoding is a tree structure that uses pointers. Each node in a quadtree is stored as a record containing six fields.

The quadtree was originally conceived as an alternative to binary array image representation with the goal of saving space by aggregation. However, more importantly, the hierarchical nature of the quadtree also results in savings in execution time. In particular, algorithms using quadtrees have execution times that are dependent not on their size but on the number of blocks in the image. Nevertheless by virtue of the tree structure, the amount of space required for pointers from a node to its children is not trivial. As a result, there has been a considerable interest in pointerless quadtree representations.

The method for representing quadtrees without pointers was developed in the form of a preorder tree

† 정 회 원:경북대학교 전자공학과
논문접수:1996년 11월 5일, 심사완료:1996년 12월 30일

traversal (i.e., depth-first order) of the nodes of the quadtree. Preorder tree traversal representations for multi-valued images have been investigated by a number of researchers such as Oliver and Wiseman[5], Woodwark[6], and Williams[7]. A preorder tree traversal representation for a binary image was proposed by Kawaguchi and Endo[8], which is called the Depth-First expression and Cohen, et al[9].

In the DF-expression the tree consists of the symbols "G", "B", and "W" corresponding to GRAY, BLACK, and WHITE nodes, respectively. The original image reconstructed from the DF-expression by observing that each nonterminal, i.e., GRAY node always has four children. While encoded DF-expression of quadtrees is more space efficient, it suffers the high cost of some operations such as the Boolean and is essentially oriented toward sequential operation on the whole quadtree such as image transmission.

In this paper we propose a new preorder tree traversal method for binary images, named the Compact Complementary Quadtree (CCQ), based on the Complementary Quadtree(CQT)[10]. In the proposed method we use type codes for representing nodes in the quadtree instead of using the symbols "G", "B", and "W". From the experimental results, we have confirmed that the CCQ has higher compression ratio than that of the DF-expression. CCQ can be effectively applied to progressive transmission of binary images such as line drawings, geographical maps, contour images extracted from gray-level images, and halftoned images.

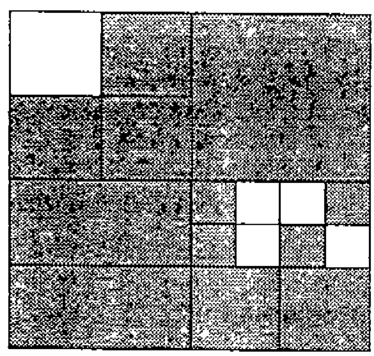
2. Reviews on the DF-expression

Quadtrees are trees in which every nonterminal node has exactly four children. In addition, the trees have labeled nodes and the children of a given node are ordered. Any given node in the quadtree represents a square subimage with sides of length 2. If this subimage is uniform in gray level then the node level indicates the gray level and this node is a terminal node of the tree. Otherwise, the label indicates that the subimage is nonuniform and four subtrees are created describing the four subblocks obtained by splitting the square into quadrants. Fig. 1 illustrates this process.

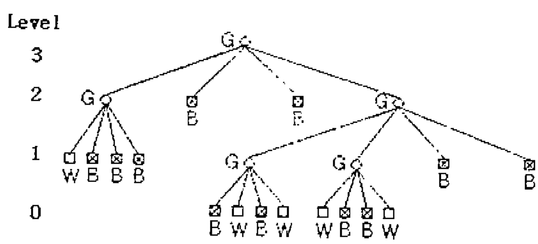
In a quadtree encoding, the tree is transmitted in a specified order (in our case, depth-first). For each node a particular binary code for the node's label is transmitted. For the tree in Fig. 1, the encoding would be in symbolic form:

```
G
G         B B G
W B B B     G           G          B B
          B W B W    W B B W
```

The vertical spacing here is for clarity only (i.e., the tree structure is still discernible); the symbols are transmitted as they appear left-to-right.



(a) A simple image



(b) Quadtree for DF-expression

(Fig. 1) A simple image and a corresponding tree for DF-expression

It is assumed that the receiver knows the size of the picture. Hence, when it encounters a "G" that cuts a 2×2 subblock into four single pixels, it knows that the next four symbols describe single pixels, and that no further cuts are possible. Thus, a block of four pixels can be represented using four bits instead of eight.

3. CCQ for Binary Images

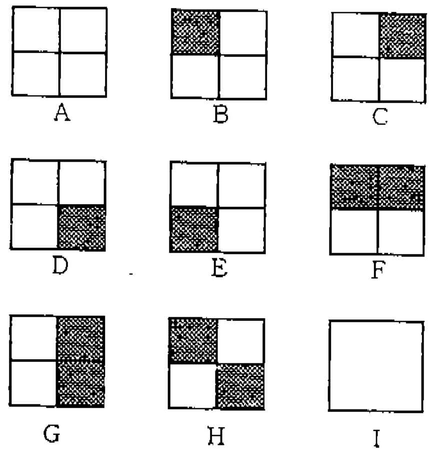
We propose a new preorder tree traversal coding method for binary images named the CCQ, based on the CQT[10]. CQT is a pointerless quadtree representation method for gray-scale images that is composed of two phases. The first phase, the color types of the gray nodes are redefined for the color which is dominant among their four children from bottom to top. The second phase, the quadcode of each node is generated only if its color is different from that of its father. CCQ for binary images is composed of two phases.

Phase 1: Redefine the color of the gray nodes from bottom to top as follows.

- If a node has three or more black children, then it is black.
- If a node has three or more white children, then it is white.
- If a node has two black and two white children, and its configuration is one of the node types shown in Fig. 2, then it is white.
- All the other nodes are black.

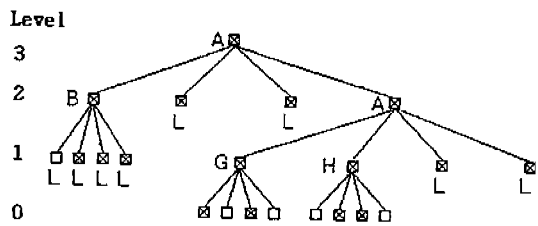
Phase 2: Determine the type codes of the redefined nodes in phase 1.

There are 8 possible configurations for children of any non-terminal node and one for terminal nodes. These 9 type codes are shown in Fig. 2. The degenerated code 'A' indicates that any or all of the children of that quad were redefined to identical colors in phase 1. Fig 3 shows the redefined color of each gray node



(Fig. 2) Type codes for binary images.

and the type codes of nodes of the tree in Fig. 1. In compact encoding, the type codes of nodes are transmitted in a specified order (breadth-first, i.e., top-to-bottom, left-to-right).



(Fig. 3) Quadtree for CCQ.

For each node, a particular binary code for the node's label is transmitted. For the image in Fig. 1, for example, the encoding would be, in symbolic form;

1, A, B L L A, L L L L G H L L,

where the first one bit indicates the color of the root whether it is Black(1) or White(0). In this case, the first one bit is 1 so the type code 'A' represents that four children are Black nodes and 'B' implies that four sons consist of one White node and three Black nodes. If the first bit is 0 the type code 'A' represents that four nodes are White and 'B' implies that four sons consist of one Black node and three White nodes. The receiver knows at any given point what node is about to be received, and what subimage will be described by that node, as a consequence of the information it has already received.

4. Compression Efficiency of CCQ

To evaluate the two coding methods, the algorithms were applied to a number of images. Their efficiencies are compared in terms of the number of bits per pixel of the original images.

We will now analyze the worst case for each of the methods. That is, we will determine the longest code length which might be generated for a given image size. For all two methods (DF, CCQ), the worst case analysis follows the same outline. A full encoding tree, with total branching until every single pixel has a corresponding node in the lowest level of the tree, always gives the longest code. For all the methods, a checkerboard with black and white alternating in all directions is an example of a worst case image. The reason that a full tree yields the worst case code is easy to prove. Given a full encoding tree, the only way to create a tree with less nodes is via a merge operation, where the children of a gray node are eliminated and the parent node becomes uniform. In all of our methods, a merge can never result in a longer code than the unmerged version. In some cases, however, it can result in a code with the same length. We now examine each method in turn.

In the case of DF-expressions, the number of nodes in a full quadtree for an image of size $2^n \times 2^n$ is

$$N_{DF} = \sum_{i=0}^{n} 2^i \times 2^i = \frac{(4^{i+1}-1)}{3} \quad (1)$$

where i represents the level of the quadtree.

In the worst case, every node in the tree requires one bit. Higher level nodes are all gray, and therefore require one bit each. The lowest level nodes also require one bit each. They require one bit instead of two bits because they are lowest level nodes (single pixels), but we restrict ourselves to images in which no other savings are possible in order to reach the worst case. Thus, (1) also gives the number of bits in the worst case image encoding.

$$B_{DF} = \sum_{i=0}^{n} 2^i \times 2^i = \frac{(4^{i+1}-1)}{3} \quad (2)$$

Dividing by the number of pixels of the original image yields a compression ratio slightly less than 4/3.

In the case of CCQ, the number of nodes except level 1 and level 0 is given by

$$N_1 = \sum_{i=0}^{n-2} 2^i \times 2^i = \frac{(4^{n-1}-1)}{3} \quad (3)$$

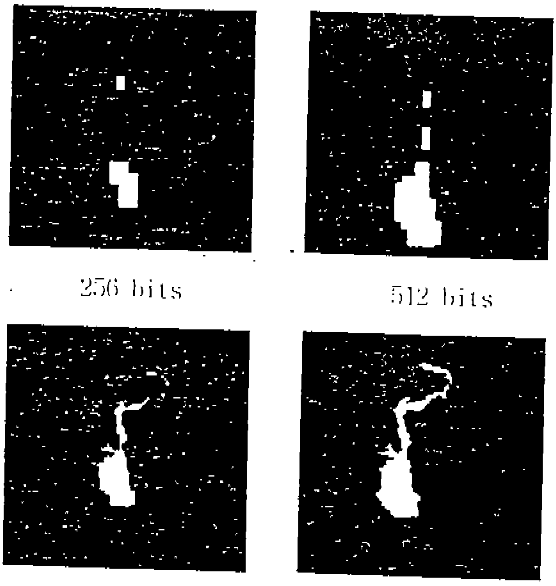
(a)

(b)

(Fig. 4) Test binary images.
(a) Floodplain. (b) Girl.

<Table 1> The results of the two methods as applied to test images.(Unit : bits)

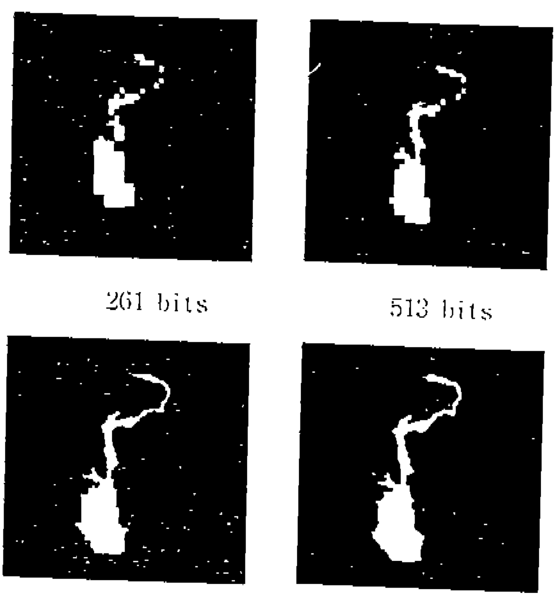
method / level	Fig. 4-a		Fig. 4-b	
	DF	CCQ	DF	CCQ
7	6	3		5 6
6	21	10	21	23
5	69	40	90	90
4	211	124	382	306
3	510	358	1347	961
2	1132	819	4084	2770
1	2374	1733	10811	7438
0	3358	3105	17919	16589



256 bits 512 bits

1025 bits 2049 bits

(Fig. 5) Progressive transmission using DF-expression.



261 bits 513 bits

1026 bits 2052 bits

(Fig. 6) Progressive transmission using CCQ.

and type code of nodes are all 'A', therefore they require 2 bits each. The number of nodes in level 1 is 4^{n-1}, and they require 4 bits each. Thus, the number of bits will be

$$B_{CC}=\frac{(4^{n-1}-1)}{3}\,2+4^{n-1}\,4 \tag{4}$$

A compression ratio is slightly less than 7/6.

5. Experimental results and conclusions.

Both DF-expression and CCQ coding methods have been tested on the images in Fig. 4-a and Fig. 4-b. Table 1 shows the results of the two methods as applied to each image. From Table 1, we know that CCQ has a higher compression ratio than that of DF-expression. The hierarchical nature of CCQ and DF-expression enables one to use progressive image transmission. Fig. 5 and Fig. 6 show progressive transmission of Fig. 4-a by use of DF-expression and CCQ. From the results, we have confirmed that CCQ has a higher compression ratio than that of DF-expression and CCQ is very effective for progressive transmission of binary images such as line drawings, geographical maps, and halftoned images.

References

[1] H.Samet and R.E. Webber, "Hierarchical data structure and algorithms for computer graphics," IEEE Computer Graphics and Applications, pp. 48-68, May, 1988.

[2] H.Samet, "The quadtree and related hierarchical data structure," ACM Computing Surveys, vol. 16, no. 2, pp. 187-260, 1984.

[3] G.M.Hunter and K. Steiglitz, "Operations on images using quadtrees," IEEE Trans. PAMI, vol.1, no.2, pp.145-153, Apr., 1979.

[4] H.Samet, "An algorithms for converting rasters to quadtree," IEEE Trans. PAMI, vol.3, no.1, pp.93-95, Jan., 1981.

[5] M.A.Oliver and N. E. Wiseman, "Operations on

quadtree encoded images," The Computer Journal, vol.26, no.1, pp.83-91, Feb., 1983.

[6] J.R.Woodwark, "Compressed quadtrees," The Computer Journal, vol.27, no.3, pp.225-229, 1984.

[7] R.Williams, "The goblin quadtree," The Computer Journal, vol.31, no.4, pp.358-363, 1988.

[8] E.Kawaguchi and T. Endo, "On a method of binary picture representation and its application to data compression," IEEE Trans. PAMI, vol.2, no.1, pp.27-35, Jan, 1980.

[9] Y.Cohen, et al, "Hierarchical coding of binary images," IEEE Trans. PAMI, vol.7, no.3, pp. 284-298, May, 1985.

[10] Y. M. Kim and S. B. Park, "Complementary quadtree," Image and Vision Computing, vol.11, no.2, pp.413-418, 1993.

김 영 모

1980년 경북대학교 전자과 졸업(학사)
1983년 KAIST 대학원 졸업(공학석사)
1989년 KAIST 대학원 졸업(공학박사)
현재 : 경북대학교 전자과 부교수
주관심분야 : 컴퓨터 그래픽스, 영상 처리

조 영 우

1992년 경북대학교 전자과 졸업(학사)
1994년 경북대학교 대학원 전자과 졸업(공학석사)
현재 : 경북대학교 전자과 박사과정 재학중
주관심분야 : 멀티미디어 시스템, 영상 처리