

二值图像的紧凑二叉树表示及其编码方法

谈国新 林宗坚

(武汉测绘科技大学遥感与信息工程学院, 武汉市珞喻路39号, 430070)

摘 要 提出一种用于图像表达的紧凑二叉树方法, 讨论了该表示方法的编码及特点, 并给出了它与线性四叉树之间的转换算法。文中所列试验及分析表明, 新提出的紧凑二叉树表示方法具有较高的时空效率。

关键词 线性四叉树; 紧凑二叉树表示; 转换算法; 时空效率

分类号 TP311.12

四叉树是一种把位于正方形区域的二维图像递归地均匀四等分的数据结构。如果一个四分区具有与目标一致的属性, 则称为黑结点; 如果一个四分区完全不包含目标, 称为白结点; 否则称作灰结点。对灰结点要继续分割, 直到所有的四分区均为黑结点或白结点为止。四叉树^[1]最初是以指针方式表示的, 每个结点需要4个指向子结点及1个指向兄弟结点的指针, 这些与位置无关的指针大大增加了存贮开销。为此, 人们采用了线性四叉树^[2]表示方法。

线性四叉树只存贮最后黑结点, 各结点间的层次关系隐含在表示结点位置的定位码 LC (Locational Code) 的顺序中。定位码常用基于四进制的 Morton 码表示。对于 $2^n \times 2^n$ 的图像, 其表达形式为:

$$LC = q_{n-1}q_{n-2}\cdots q_{n-i}\cdots q_0 \quad (1)$$

式中 q_{n-i} 是第 i 次分割时目标所在的四分区码。

从(1)式可以看出, 定位码由 n 位四进制数构成。当 n 较大时, 其所占空间相当可观。为避免定位码的存贮, 也可对四叉树中所有的叶结点前序遍历按 (LV, GR) 的格式存贮。式中 GR (Grey) 代表结点的灰度值, LV (Level) 为层次码。此时叶结点的定位码隐含在第一个叶结点到当前结点的长度 (像素个数) 中。

本文根据四叉树结构的特点, 提出了一种新的紧凑二叉树表示方法。该方法除沿袭了四叉树的层次特征外, 还能有效地提高存贮效率。

1 紧凑二叉树表示方法及编码特点

在线性四叉树中, 叶结点定位码的前缀部分有许多数字是相同的。如果把各定位码中的每一位数字分别在相应结点中存贮, 则可达到进一步压缩空间的目的。不过, 这样表示必须在存贮黑结点信息的同时也存贮有关灰结点的信息。由于四叉树结构所需指针较多, 因此要真正达到压缩空间的目的, 还需把四叉树结构化简成二叉树形式。

用二叉树形式表示四叉树结构, 每个结点可由3部分构成: 指向下一层结点的指针 SP (Son Pointer); 由四进制数表示的四分区码 QN (Quadrant Number); 指向同层后续结点的指针 BP

(Brother Pointer)。图1表示了各结点的存贮内容及其简化表示形式。

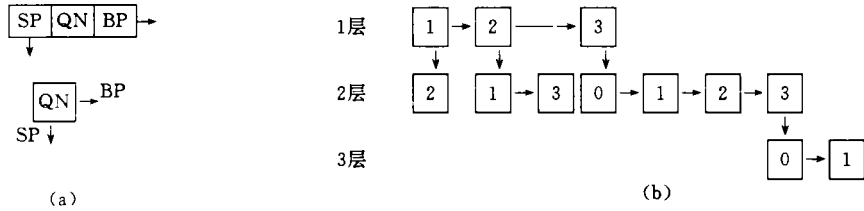


图1 四叉树结构的二叉树表示

图1b 所示的二叉树是四叉树结构的简化表示,对原四叉树结构的各种搜索也很容易在当前二叉树结构中实现。例如,具有同一父结点的各子结点可通过 BP 指针逐一访问,而不同层次的结点则通过 SP 指针访问,其中第一个结点属于第一层,以后每沿 SP 指针向下访问一次,其层次加1。如果 SP 指针为空,则该结点即为相应四叉树结构中的一个黑结点。而从第一层出发沿 SP 指针到达黑结点所经过的各层四分区码组成的序列,就是该结点的定位码。

显然,要进一步压缩存贮空间,必须丢弃二叉树结构中的所有指针。为此,这里采用了一种紧凑二叉树表示方法 CBR(Compact Bintree Representation)。该方法直接对图1b 所示的二叉树结构按前序遍历进行扫描,从而得到紧凑的代码序列。序列中的每个代码对应于不同的结点,且其存贮内容由三部分构成:四分区码 QN;子结点标志 ST;兄弟结点标志 BT,其中 ST 和 BT 都是一个二进制标志。当 ST 为1时,表示该结点是灰结点,即存在子结点;否则表示黑结点。如果 BT 的值为1,表示该结点有后续兄弟结点;否则说明它是同层中的最后一个结点。按 (QN,ST,BT)格式编码方法对图1b 进行前序遍历,可形成下列 CBR 代码序列:(1,1,1)(2,0,0)(2,1,1)(1,0,1)(3,0,0)(3,1,0)(0,0,1)(1,0,1)(2,0,1)(3,1,0)(0,0,1)(1,0,0)。

上述 CBR 代码序列具有以下特点:

- (1) 每个代码的空间开销为4bit,一个字节正好存贮两个结点的信息,因此 CBR 方法便于按字节存贮和处理。
- (2) 序列中的每个代码相应于四叉树结构中的一个结点。ST 值为1表示对应的是灰结点,其后续代码就代表该灰结点的某个子结点;ST 值为0,则表示它对应于四叉树结构中的某个黑结点。
- (3) 序列中 BT 值为1的代码,说明它所对应的四叉树结点有后续兄弟结点,且其兄弟结点的代码位置可这样确定:如果当前代码的 ST 值为0,则该代码的后续代码就是要找的兄弟结点代码;否则定义一变量的值为0,并从当前代码开始顺序扫描序列。当代码中的 ST 和 BT 值同时为1时,变量加1;而当代码中的 ST 和 BT 值同时为0时变量减1,并判断变量值是否为0。如是,则停止扫描,其后续代码就是所找的兄弟结点代码。
- (4) 利用 CBR 序列也能方便地进行各种四叉树结点查询。其过程为:由序列的第一个代码开始沿 ST 标志向各层搜索,在各层中把第一个子结点对应代码的 QN 值与给定线性四叉树结点相应数字相比较。如 QN 值大,则查找失败;如相等,就沿该代码的 ST 标志进行下一层的搜索;如 QN 值小,则再与该代码的兄弟结点代码比较,若无后续兄弟结点,查找失败。
- (5) CBR 序列也可体现四叉树结构的合并与分裂操作。例如,若连续有四个代码的 ST 值为0,且前三个代码的 BT 值均为1,则可删除这些代码,并把它们父结点代码的 BT 值改成1。
- (6) 顺序扫描一次序列,可得线性四叉树所有的信息,即用序列可恢复原图像。

2 CBR 与线性二叉树之间的转换

2.1 线性二叉树到 CBR 的转换

文献[3]介绍了一种线性二叉树的快速构造算法。该算法输出的线性二叉树叶结点是按(LV,GR)格式存贮的,因此在线性二叉树与 CBR 的转换算法中,也假设线性二叉树按这一格式输入或输出。由于二叉树结构中灰结点信息隐含在线性二叉树序列中,而 CBR 序列中却要显式表示,因此为实现线性二叉树到 CBR 的转换,先定义一个表示灰结点层次及其四分码大小的数组 Lab [n], 以及一个指示当前层次的变量 LP。其中数组 Lab [] 的下标表示层次,其值表示该层次当前结点的四分码。然后,把线性二叉树转换成结构为(LV, QN, ST)的中间缓冲序列,最后再把中间缓冲序列转化为 CBR 序列。其转换算法概括如下:

- (1) 用3初始化数组 Lab [], 并给当前层变量 LP 赋值1。
- (2) 自前往后扫描输入线性二叉树叶结点值(LV, GR)。若 LV=LP, 转(4); 否则转(3)。
- (3) I 从 LP 取值到 LV-1, 若 Lab [I]=3, 则赋 Lab [I]=0, 输出(I, 0, 1); 否则令 Lab [I]=Lab [I]+1, 且输出(I, Lab [I], 1)。
- (4) 先判断 Lab [LV] 的值, 如为3则赋0; 否则 Lab [LV]=Lab [LV]+1。然后判断 GR 的值是否为1, 如是则输出(LV, Lab [LV], 0)。最后 LP=LV。
- (5) 若 Lab [LP]=3, 则取 LP=LP-1, 重复步骤(5); 否则转(2), 直到所有的叶结点输入完毕为止。
- (6) 由前往后顺序扫描输入已生成的中间缓冲序列值(LV, QN, ST)。
- (7) 逐个与后续中间缓冲序列值(LV', QN', ST') 比较, 如果 LV' ≤ LV, 转(8); 否则重复(7), 直到最后一个序列值, 此时输出(QN, ST, 0)后转(9)。
- (8) 若 LV'=LV, 则输出(QN, ST, 1); 否则输出(QN, ST, 0)。
- (9) 转(6)直到最后一个中间缓冲序列值。

2.2 CBR 到线性二叉树的转换

定义一个用来指示当前层次的变量 LP, 以及一个结构为(Q, B)的一维数组 Lab [n], 用于存贮转换过程中各层的有关信息。整个转换算法可概括如下:

- (1) 以(0, 0)初始化 Lab [] 数组, 且令 LP=1。
- (2) 由前往后逐一扫描输入(QN, ST, BT)。如果 QN=Lab [LP]·Q, 转(4); 否则转(3)。
- (3) 输出 QN-Lab [LP]·Q 个线性二叉树叶结点(LP, 0)。
- (4) 置 Lab [LP]·Q=QN+1, Lab [LP]·B=BT。如果 ST=0, 输出(LP, 1)后转(5); 否则 LP=LP+1, 转(2)。
- (5) 如果 Lab [LP]·B=0, 转(6); 否则转(2)。
- (6) 如 Lab [LP]·Q=4, 则令 Lab [LP]·Q=0, LP=LP-1; 否则输出4-Lab [LP]·Q 个(LP, 0), 并置 Lab [LP]·Q=0, LP=LP-1。若 LP=0, 结束返回; 否则转(5)。

3 时空效率分析及试验

3.1 时空效率分析

设 $2^n \times 2^n$ 的图像中只包含一个 $2^m \times 2^m$ (常取 $m = n - 1$) 的正方形区域目标, 此正方形

目标位于不同的位置可形成不同的四叉树。分最好与最坏两种情况分别对 CBR 和线性四叉树的 存贮效率进行分析比较,其中线性四叉树又包括(LC, LV)和(LV, GR)两种结构,分别称作 LQ1法和 LQ2法。

最好的情况是整个区域目标在图像中只用一个黑结点表示,此时 LQ1、LQ2 和 CBR 法的空间开销分别为 $2m + 2 + \lceil \log_2(m + 1) \rceil \text{bit}$, $4 + 4 \lceil \log_2(m + 1) \rceil \text{bit}$ 和 4bit , 式中符号“ \lceil ”表示舍入取整。显然, CBR 法存贮效率优于其它两种方法。

最坏的情况是最好情况下的目标向图像中心上下左右各移动一个像素,如图2所示。在这种情况下,四叉树灰结点个数 G 、黑结点个数 B 和叶结点数 L 的表达式为:

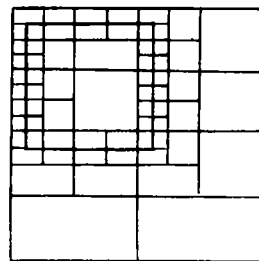


图2 最坏情况

$$G = m + 2(2^m - 1) + \sum_{i=2}^m (1 + 2^{m+2-i} - 2) + 2 = 2^{m+2} - 3 \tag{2}$$

$$B = (2^{m+1} - 1) + \sum_{i=0}^{m-1} (2^{m-i+1} - 3) = 3 \cdot 2^{m+1} - 3m - 5 \tag{3}$$

$$L = 3(2^{m+2} - 3) + 1 = 3 \cdot 2^{m+2} - 8 \tag{4}$$

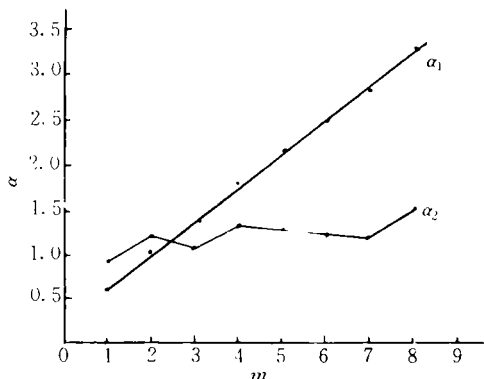
由(2)、(3)两式相加可得 CBR 结点数 C :

$$C = 5 \cdot 2^{m+1} - 3m - 8 \tag{5}$$

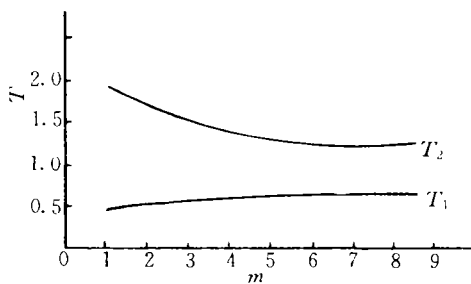
显然 $C > B$ 。但由于线性四叉树存贮结点的字长随 n 的增大而增加,而 CBR 结点存贮字长固定为 4bit 。因此在某个 n 值下, CBR 可使代码总长压缩。由(3)、(4)和(5)式可得 CBR 相对于 LQ1 和 LQ2 所提高的压缩倍数 α_1 和 α_2 :

$$\alpha_1 = \frac{\text{LQ1}}{\text{CBR}} = \frac{(3 \cdot 2^{m+1} - 3m - 5)[2(m + 1) + \lceil \log_2(m + 1) \rceil]}{5 \cdot 2^{m+3} - 12m - 32} \tag{6}$$

$$\alpha_2 = \frac{\text{LQ2}}{\text{CBR}} = \frac{(3 \cdot 2^{m+2} - 8) \cdot [1 + \lceil \log_2(m + 1) \rceil]}{5 \cdot 2^{m+3} - 12m - 32} \tag{7}$$



(a)



(b)

图3 压缩比 α_1 、 α_2 和结点数比 T_1 、 T_2 随 m 变化情况

图3a给出了(6)式和(7)式随 m 变化的曲线。从图中可以看出,压缩倍数 α_1 随 m 的增大而线性增加,且其提高幅度较大。 α_2 随 m 的增大跳跃增加,当 m 为 2 的整幂次方时, α_2 值剧增,然

后随 m 值增大而略微减少,直到 m 值为 2 的整幂次方为止。从整个曲线走势来看, α_2 曲线仍是逐步上升的。当 $m > 1$ 后, CBR 比 LQ1 和 LQ2 所需的存贮空间都少。当 $m = 8$ 时, CBR 存贮效率是 LQ1 的 3.3 倍, 是 LQ2 的 1.5 倍。

利用线性二叉树能直接进行各种图形操作及代数运算, 其时间复杂度与参加运算的结点总数成线性关系^[5]。由于 CBR 和线性二叉树都属二叉树代码序列表达技术, 因此上述结论同样适用于 CBR。由 (3)、(4) 和 (5) 式可知, CBR 与 LQ1 和 LQ2 之间的时间效率关系可用它们之间的结点数比 $T_1 = \frac{B}{C}$ 和 $T_2 = \frac{L}{C}$ 表示。图 3b 给出了 T_1 和 T_2 随 m 值变化的曲线。

从图 3b 可以看出, T_1 是一递增曲线, T_2 为递减曲线, 其递增(减)速度随 m 的增大而减小。当 $m \geq 7$ 时, T_1 和 T_2 值变化不大, 分别收敛于 0.6 和 1.2 左右, 说明 CBR 的时间效率恒高于 LQ2, 但不及 LQ1。与图 3a 综合看, CBR 在时间效率和存贮效率方面均优于 LQ2; LQ1 虽然在时间效率方面优于 CBR, 但随着 m 的增大, 不仅其存贮效率急剧下降, 而且时间效率也逐步回落, 说明 CBR 比 LQ1 更适用于高分辨率 ($m \geq 8$) 图像的表达。

3.2 试验

为验证上述分析结果, 这里任意选取了 4 幅大小为 $2^9 \times 2^9$ 的二值区域图像, 分别用 CBR、LQ1 和 LQ2 等方式表达这些图像, 其结点数及空间开销列于表 1。

表1 不同表达方法的结点数及空间开销(/bit)

| | image1 | | image2 | | image3 | | image4 | |
|-----|--------|--------|--------|--------|--------|--------|--------|--------|
| | 结点数 | 空间开销 | 结点数 | 空间开销 | 结点数 | 空间开销 | 结点数 | 空间开销 |
| CBR | 33 961 | 16 981 | 27 984 | 13 992 | 22 006 | 11 003 | 16 470 | 8 235 |
| LQ1 | 19 861 | 54 626 | 16 203 | 44 561 | 12 590 | 34 617 | 9 239 | 25 400 |
| LQ2 | 42 304 | 26 440 | 35 344 | 22 090 | 28 252 | 17 663 | 21 697 | 13 560 |

从表中可以看出, CBR 的存贮效率约比 LQ1 高 3.2 倍, 时间效率则为 LQ1 的 58% 左右; 与 LQ2 相比, CBR 的存贮效率和时间效率分别高约 1.6 倍和 1.28 倍。显然, 试验结果与前面分析相吻合。

值得注意的是, 图 3 和表 1 中 LQ1、LQ2 两种表达方法的空间开销均按比特存贮和计算。如果按整字节存贮计算, 则 LQ1 和 LQ2 表达方法的空间开销都会有不同程度的增加。而 CBR 方法每个结点的空间开销固定为 4bit, 不随图像大小的变化而变化, 且 1bit 正好存贮两个结点, 从而便于按字节处理。

参 考 文 献

- 1 Samet H. The Quadtree and Related Hierarchical Data Structures. ACM Comput. Surv., 1984, 16(2): 187~260
- 2 Gargantini I. An Effective Way to Represent Quadtrees. ACM Commun., 1982, 25(12): 905~910
- 3 谈国新, 林宗坚. 基于自然数的线性二叉树优化构造算法. 测绘学报, 1995, 24(3): 204~210
- 4 Dyer C. The Space Efficiency of Quadtrees. Comput. Graphics Image Process, 1982, 19(4): 348~355
- 5 Bauer M A. Set Operations on Linear Quadtrees. Computer Vision Graphics and Image Processing, 1985 (29): 214~223

靠电池供电,程序运行的机时增加,就意味着一次充电的使用时间减少,这对某些应用场合极为不利。若划分得太大,又会使“碎片”问题严重。在开创文件数目较多时,这种“碎片”所占的内存量也相当可观,使本来就紧张的存储空间雪上加霜。所以,应根据具体的硬件和使用情况酌情加以考虑。

HEB₃电子手簿的应用表明,本文的文件管理系统,其设计方法是合理且有效的,并完全实现了最初的设计思想。若将其管理体制运用于其它系统,只需对 OPMEM 程序进行改动,其余程序均可不动或不作大的改动,这充分表明了其移植性强的特点。

参 考 文 献

- 1 张载鸿. 局部网操作系统 DOS 高技术分析. 北京:国防工业出版社,1988.
- 2 何立民. 单片机应用系统设计. 北京:北京航空航天大学出版社,1990.

A Design of File Management System in Electronic Fieldbook

He Jianying Yuan Yuzheng

(School of Electronic Information Engineering & Precision Mechanics, WTUSM, 39 Luoyu Road, Wuhan, China, 430070)

Abstract This paper offers a method of file management in electronic fieldbook, describes some related problems, such as how to establish the file directory table and the file assignment table, and how to assign memory. The described method has been used in HEB3A electronic fieldbook.

Key words electronic fieldbook; file management; single - chip processor application



(上接223页)

The Compact Bintree Representation & Encoding of Binary Images

Tan Guoxin Lin Zongjian

(School of Remote Sensing and Informatics, WTUSM, 39 Luoyu Road, Wuhan, China, 430070)

Abstract In this paper, the authors propose a method of compact bintree representation (CBR) used for binary images, and discuss its encoding characteristics. Then, the conversion algorithm between CBR and linear quadtree is given. The comparison and examples in this paper show that the spatiotemporal efficiency of CBR is higher than linear quadtree's.

Key words linear quadtree; compact bintree representation; conversion algorithm; spatiotemporal efficiency