环 境 遥 感
第6卷 第4期 REMOTE SENSING OF ENVIRONMENT Vol. 6, No. 4
1991 年 11 月 CHINA Nov., 1991

遥感图像栅格—2DRE 四叉树结构变换算法

萧 柯
(中国科学院遥感应用研究所)

1990 年 10 月 6 日收稿

摘 要

本文提出并分析了一个栅格—四叉树结构间的变换算法。栅格和四叉树这两种结构各适用于某些特定的运算,各有其优、缺点,能够互相补充。栅格结构是最常见的图像数据结构,而四叉树是近年来才得到广泛的注意和应用,从而有了很大发展的一种新的数据结构。在本文中,简述了四叉树的发展,并对变换算法的基础及算法作了详细的阐述,进而对其进行了评价和分析。

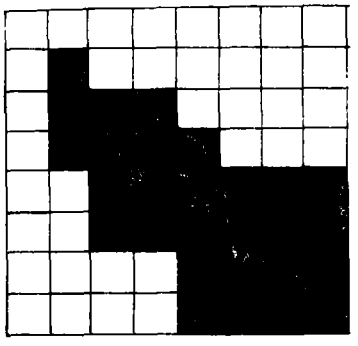
关键词 遥感 图像处理 数据结构 四叉树 地理信息系统

一、引 言

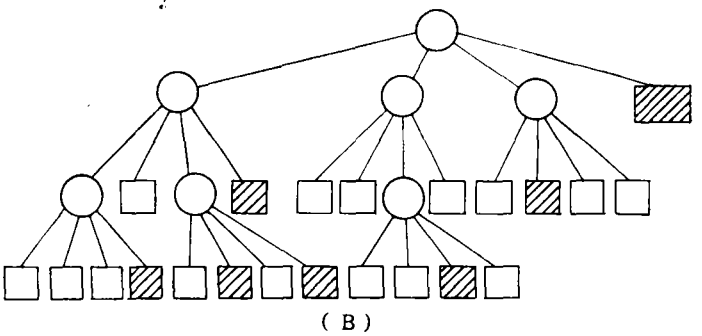
图像数据结构的研究,随着图像、图形处理以及地理信息系统等应用领域的发展,越来越受到了人们的重视。其中,栅格 (Raster) 和四叉树 (Quadtree) 是比较重要的两种结构。

栅格结构是最常见的一种图像数据结构,它将图像表示为一个矩阵的形式,具有自然、直观、方便等优点。图像的最初获得,经常是通过扫描设备,而最终的输出,通常也要通过行输出的显示设备,这就决定了这种数据结构的重要地位。

四叉树是最近十余年来发展起来的一种数据结构。1976 年, A. Klinger 和 C. R. Dyer 等人为了对复杂图像建立一种计算机可检索的数据表示法,提出了"均匀分割"的概念,并定义了相应的分割方法[1]。按照这一方法,对一幅图像进行连续的均匀四分,其结果就是一棵四叉树 (图 1)。它具有结构清晰、简单、规整且便于检索等特点。但因这种树结构延袭了传统的树的链式结构,含有大量的冗余信息 (图 2)。所以,它所耗费的存储空间和 I/O 所费时间不可避免地大到惊人的程度。于是, 1982 年, I. Gargantini 提出了一种称为"线性四叉树" (Linear Quadtree) 的新的结构形式[2]。在这种结构中,由于仅存储结点,且对每一结点进行编码,而这种编码本身就隐含了从根到结点的路径 (图 3),这就使人们得以摆脱传统的树结构中链、指针以及中间结点等冗余信息所带来的耗费大量存储空间和 I/O 时间的困扰。1985 年, J. P. Lauzon 等人又提出了一种新的线性四叉树的编码方式[3]。他们称之为"二维行程编码" (2 Dimensional Run-Encoding 简称 2DRE)。这种方法采用所谓 "Morton" 序列进行编码,同时,对于线性四叉树中结点的相邻者进行归并,以得到最大的连续序列 (图 4)。从而在节省存储空间方面又进了一步。



(A)



(B)

图 1 图像及其四叉树

(a) 一幅 8×8 的二值图像

(b) 由(a)均匀分割得到的四叉树

Fig. 1 An Image and Its Quadtree

(a) A 8×8 Binary Image

(b) A Quadtree from Regular Decompositing the Image (a)

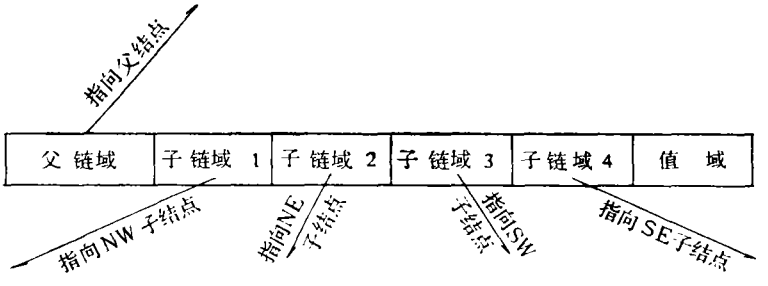


图 2 传统的四叉树中的结点

Fig. 2 A Node in Tridional Quadtrees

在进行理论研究的同时，不少学者对于如何在实际中具体应用四叉树结构的问题也进行了若干研究与探索，特别是在地理信息系统方面。如 1983 年 D. J. Abel 等人研究了线性四叉树应用于地理数据库的矩形区域的检索问题[4]。 1984 年，D. J. Abel 又提

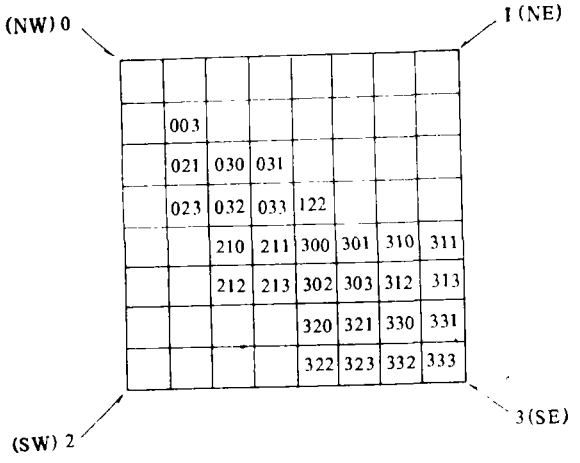


图 3　图 1(a) 的线性四叉树编码

Fig. 3　Linear Quadtree Encoding of Fig. 1(a)

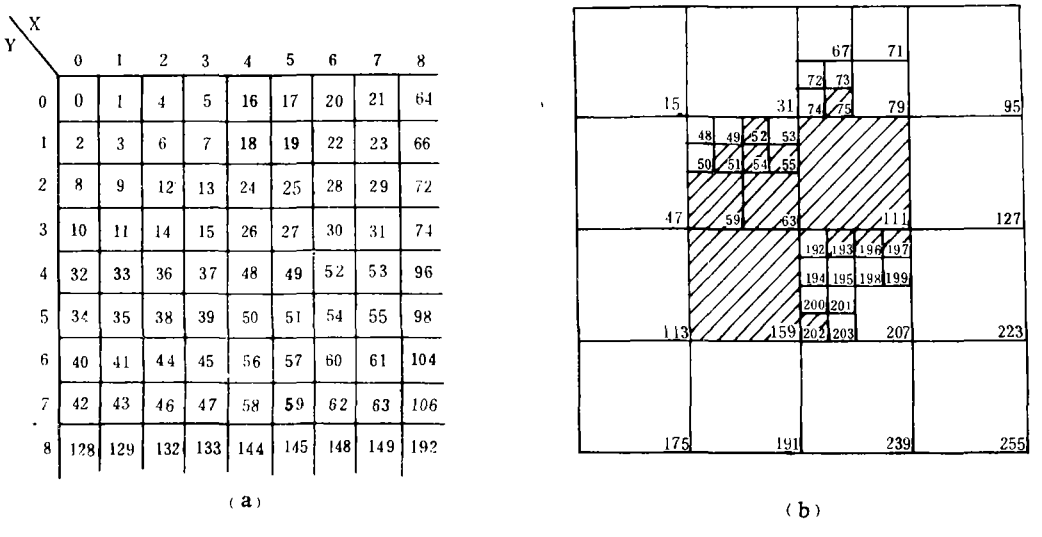


图 4　2DRE 线性四叉树的编码方式

(a) 用 Morton 序列编码的一幅图像的前 64 个像元及其邻接像元

(b) 用 Morton 序列编码的一幅 16×16 的二值图像

Fig. 4　Encoding of 2DRE Linear Quadtrees

(a) First 64 Pixels and Their Adjacent Pixels of an Image in Morton Serial

(b) A 16 × 16 Binary Image in Morton Serial

出用 B^+ 树来表示大四叉树，以便将其用于地理数据库[5]。1986 年，F. D. Libera 和 F. Gosen 提出用 B-树结构来解决地理信息的查询问题[6]。最近，又有一些文章讨论了四叉树及其在地理信息系统中的应用问题[7,8]。

从栅格和四叉树这两种结构的比较中不难看出，对于图像的显示和输出，栅格结构无疑是最为合适的，但也具有一些固有的缺点：它不易表达图像的结构信息，在进行某些处

理、运算时有若干冗余信息，从而要花费较多的时间、空间等等。而四叉树结构恰恰在这些方面补其不足，特别是 2DRE 四叉树在压缩容量，节省空间方面有着明显的优点，而缺点可由栅格结构加以补足。在一个较为完善的图像处理系统或地理信息系统中，应充分利用不同数据结构各自的特点，以满足不同的要求，从而提高系统对图像数据进行存储、处理、查询的时间、空间效率。而这方面的研究，目前还比较少。针对这种情况和实际要求，本文提出了一个在栅格和 2DRE 四叉树结构之间进行变换的算法，即 Raster to 2DRE Quadtree变换算法。在以下各节中，将对其作详细描述，介绍算法的基础，即由座标值到二维行程编码的转换；再叙述算法；然后对算法进行分析和评价。

二、编 码 转 换

设有一幅 $2^n \times 2^n$ 的数字图像，$[i,j]$ 表示图像中任意一点 P 的 x，y 坐标，则有：

$$i = x_{n-1} \cdot 2^{n-1} + x_{n-2} \cdot 2^{n-2} + \cdots + x_1 \cdot 2^1 + x_0 \cdot 2^0 \tag{1}$$

$$j = y_{n-1} \cdot 2^{n-1} + y_{n-2} \cdot 2^{n-2} + \cdots + y_1 \cdot 2^1 + y_0 \cdot 2^0 \tag{2}$$

其中 i，j 均以十进制数表示，$x_l, y_l \in \{0,1\}$，$l = 0,1,\cdots,n-1$。令 $Q(i,j)$ 为 $x = i$，$y = j$ 处的四叉树二维行程编码，根据这种编码的构成规则，有：

$$\begin{aligned} Q(i,j) &= y_{n-1} \cdot 2^{2(n-1)+1} + y_{n-2} \cdot 2^{2(n-2)+1} + \cdots + y_2 \cdot 2^5 + y_1 \cdot 2^3 + y_0 \cdot 2^1 + \\ &\quad x_{n-1} \cdot 2^{2(n-1)} + x_{n-2} \cdot 2^{2(n-2)} + \cdots + x_2 \cdot 2^4 + x_1 \cdot 2^2 + x_0 \cdot 2^0 \\ &= \sum_{l=0}^{n-1} y_l \cdot 2^{2l+1} + \sum_{l=0}^{n-1} x_l \cdot 2^{2l} \end{aligned} \tag{3}$$

显然

$$Q(i,j) = Q(i,0) + Q(0,j) \tag{4}$$

当 $i = j$ 时又有：

$$\begin{aligned} Q(0,i) &= 2 \cdot Q(i,0) \\ Q(0,j) &= 2 \cdot Q(j,0) \end{aligned} \tag{5}$$

于是，根据(4)和(5)，则有：

$$Q(i,j) = Q(i,0) + 2Q(j,0) \tag{6}$$

由此，只要得到图像第一行（y 坐标 $j = 0$）中每一点的编码值，其余各行中任意点的编码值即可依次获得。由(3)得：

$$Q(i,0) = x_{n-1} \cdot 2^{2(n-1)} + \cdots + x_1 \cdot 2^2 + x_0 \cdot 2^0 \tag{7}$$

其中：

$$\begin{aligned} x_0 &= i \bmod 2 \\ x_1 &= \left\lfloor \frac{i}{2} \right\rfloor \bmod 2 \\ x_2 &= \left\lfloor \frac{\left\lfloor \frac{i}{2} \right\rfloor}{2} \right\rfloor \bmod 2 \\ &\cdots\cdots \end{aligned} \tag{8}$$

$$x_{n-1}=\left\lfloor\frac{\overbrace{\left\lfloor\frac{\left\lfloor\frac{i}{2}\right\rfloor}{2}\right\rfloor\cdots\cdots\left\lfloor\frac{\left\lfloor\frac{i}{2}\right\rfloor}{2}\right\rfloor}^{n-1\text{重}}}{2}\right\rfloor \bmod 2$$

写成递推形式：

$$\begin{aligned}T_0&=1\\T_1&=\left\lfloor\frac{i}{2}\right\rfloor\\T_2&=\left\lfloor\frac{T_1}{2}\right\rfloor\\&\cdots\cdots\\T_{m+1}&=\left\lfloor\frac{T_m}{2}\right\rfloor\end{aligned}\tag{9}$$

于是又有：

$$\begin{aligned}T_0-2T_1&=x_0\\T_1-2T_2&=x_1\\&\cdots\cdots\\T_{n-1}-2T_n&=x_{n-1}\end{aligned}\tag{10}$$

求得了 $x_l(l=0,1,\cdots,n-1)$，即得到 $Q(i,0)$（据(7)式）。令 i（即 x 的坐标值自 0 变换到 2^n-1，则依次得到图像中第一行各点之编码值，再根据(6)式，即可得到图像中全部点的编码值。这样，我们就完成了从 x，y 坐标到四叉树的二维行程编码的转换。

至于从行程编码到 x，y 坐标的反转换，我们有以下分析：

根据(1)、(2)式，只要求得其中每一 x_l、y_l，便可得到 x 的坐标值 i 和 y 的坐标值 j。

令 $Q(i,j)$ 为 $x=i$，$y=j$ 处的行程编码值（以下简写为 Q）。可根据以下诸式求出 x_l，y_l，再由它们各自的权得到 i 和 j 的值：

$$\begin{aligned}K_0&=Q\\K_{\frac{1}{2}}&=\left\lfloor\frac{Q}{2^{2n-1}}\right\rfloor\\K_1&=K_0-K_{\frac{1}{2}}\cdot 2^{2n-1}\\&\cdots\cdots\\K_{m+\frac{1}{2}}&=\left\lfloor\frac{K_m}{2^{2n-(m+1)}}\right\rfloor\end{aligned}\tag{11}$$

$$K_{m+1}=K_m-K_{m+\frac{1}{2}}\cdot 2^{2n-(m+1)}\tag{12}$$

不难推得：

$$\begin{aligned}K_{\frac{1}{2}}&=y_{n-1},\\K_{1\frac{1}{2}}&=x_{n-1},\\K_{2\frac{1}{2}}&=y_{n-2},\\K_{3\frac{1}{2}}&=x_{n-2},\\&\cdots\cdots\\K_{(2l-2)+\frac{1}{2}}&=y_{n-l}\qquad l=1,2,\cdots,n\end{aligned}\tag{13}$$

$$K_{(2l-1)+\frac{1}{2}}=x_{n-l}\tag{14}$$

求得了诸 $x_m, y_m (m = 0,1,2,\cdots n-1)$，即可根据(1)、(2)式得到 $Q(i,j)$ 的 x, y 坐标 i 和 j 的十进制表示。

显而易见，坐标与其编码之间的关系是一一对应的。换言之，图像中任意一点，其编码由其 x, y 的坐标值唯一确定；反之，其坐标值亦由其编码唯一确定。据此，即可设计一种算法，进行坐标-编码之间的转换，并将其作为 Raster-2DRE Quadtree 变换算法的一个重要组成部分。

三、算法描述

根据上述的编码转换方法，即可将栅格结构的图像变换为线性四叉树的 2DRE 形式。它对图像进行处理的方式是自第一行（y 坐标为 0）开始，逐行读入、变换、归并，并且根据四叉树的结构性质，在 $N \cdot 2^i$ 的相应行处进行层次归并，保证每处理完图像的一行之后，能够得到当时最紧凑的 2DRE 四叉树，从而有效地利用存储空间，避免了整幅图像有时不能同时装入内存的矛盾。另外，还可以防止先建树，后归并的处理方法可能造成的"溢出"情况。以下简述本算法对其进行处理的过程。设有一幅 $2^n \times 2^n$ 的二值图像，其具体步骤如下：

（一）根据(7)和(8)式求出第一行中各元的 2DRE 编码值 $Q(i,0)$ $(i = 0,1,\cdots, 2^n-1)$，以作为编码基值表。

（二）置行计数器 j 的初值为 0。

（三）置层次指示器 $l(N) = N$ $(N = 1,2,\cdots,n)$。

（四）置层次指针 $K = 0$。

（五）读入图像的一行。

（六）根据行号和(6)式，利用编码基值表，得到本行中各元的 2DRE 编码值 $Q(i,j)$ $(i = 0,1,\cdots,2^n-1)$。

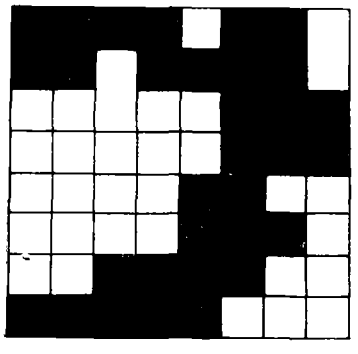


图 5 一幅 $2^3 \times 2^3$ 图像

Fig. 5 A $2^3 \times 2^3$ Image

（七）对本行中的各元逐一判别。若为 0 值者则跳过；若为 1 值者则将其编码值记入树区。

（八）将树区中属于本行的编码邻接的像元归并。

（九）将行计数器加 1$(j \leftarrow j+1)$。

（十）将层次指针加 1$(K \leftarrow K+1)$。

（十一）判别是否 $K \geqslant n$，是则转向第(十五)步；否则往下进行。

（十二）判别 j 是否满足 $j \bmod 2^{l(K)}$。是则往下进行；否则转向第(四)步。

（十三）将树区中属于本层次的编码邻接的像元进行归并。

（十四）转向第(十)步。

（十五）将得的 2DRE 编码的线性四叉树文件输出。

在本算法中，采用二维数组(设为 ARRAY $[M,2]$)，来记录四叉树。其中 ARRAY

[m,1] ($m = 1,2,\cdots,M$) 记录图像中"1"值点构成的连续区域的起始点编码；ARRAY [m,2] 则记录相应各区域的线性长度即像元数。

以下用图表来进一步说明本算法：

表 1　第一行各元的编码

Table 1　Encoding of the Pixels in First Row of the Image

0	1	4	5	16	17	20	21	

现以图 5 所示图像为例，它的第一行各元的编码列于表 1。表 2、表 3 为图像的第一行处理、归并后得到的当时 2DRE 线性四叉树结构编码，其中表 2 为第一行处理完后的树区结构编码，表 3 为第一行归并完后的树区结构编码。

表 2　第一行处理后的树区结构编码

Table 2　Tree Area of the Image with First Row Processed

0	1	4	5	17	20				
1	1	1	1	1	1				

表 3　第一行归并完后的树区结构编码

Table 3 Tree Area of the Image with First Row Merged

0	4	17	20				
2	2	1	1				

树区中的第一行内各数是编码连续区域的首元编码，第二行则是相应于第一行的各区域的值点数，以下各表同。

表 4，表 5，表 6 为图像的第二行归并后以及第一个 2^l 层次处理后的 2 DRE 四叉树结构编码。

表 4　第二行处理、归并后的树区编码

Table 4 Tree Area with the Second Row Processed and Merged

0	4	17	20	2	7	18	22		
2	2	1	1	2	1	2	1		

表 5　第一个 2^l 层次 ($l = 1$) 内插值后的树区编码

Table 5　Tree Area with the First 2^l Level Interpolated ($l = 1$)

0	2	4	7	17	18	20	22		
2	2	2	1	1	2	1	1		

表 6　2^l 层次 ($l = 1$) 经过归并后的树区编码

Table 6 Tree Area with the First 2^l Level Merged ($l = 1$)

0	7	17	22					
6	1	4	1					

表 7 为图像的第四行处理、归并后，并经过第二个 2^1 层次的插入、归并，再经过第一

个 2^2 层次内的插入，归并以后得到的当时 2DRE 四叉树的结构编码。

表 7 第四行处理、归并后四叉树结构编码

Table 7 The 2DRE Quadtree with the Second 2^1 Level Interpolated and Merged, then the First 2^2 Level Interpolated and Merged

0	7	17	22	25	27					
6	1	4	1	1	5					

表 8 第六行处理、归并后树区的编码

Table 8 Tree Area with the 6th Row Processed and Merged, then, the Third 2^1 Level Interpolated and Merged

0	7	17	22	25	27	48	54			
6	1	4	1	1	5	4	1			

表 8 为图像的第六行处理、归并后，并经过第三个 2^1 层次的插入、归并后的树区结构编码。

表 9 第八行处理后的树区编码

Table 9 Tree Area with the 8th Row Processed and Merged, the 4th 2^2 Level Interpolated and Merged

0	7	17	22	25	27	48	54	42	56	
6	1	4	1	1	5	4	1	6	3	

表 10 经第二个 2^2 层次插入、归并后的编码

Table 10 The Final Result --- Tree Area after the Second 2^2 Level Interpolated and Merged

0	7	17	22	25	27	42	54	56		
6	1	4	1	1	5	10	1	3		

表 9，为图像的第八行处理、归并后，并经过第四个 2^1 层次的插入、归并后的树区；表 10 为最终得到的 2DRE 四叉树结构编码。

经过一系列的判别、插入、归并，得到如表 10 所示的最终结果。在整个过程中，由于采用了随时归并的策略，使每一步得到的当时的 2DRE 四叉树都能保证紧凑，这样就使空间的利用更加有效。

四、算法的评价与分析

本算法是通过对二值图像进行变换来描述的。我们在实际应用中，用其对经过分类得到的多灰度值图像进行变换，对每一灰度值均得到其相应的 2DRE 四叉树。其结果是大大压缩了原图象所占据的存储空间，这一效益正是 2DRE 四叉树的“连续区域归并”的原则带来的，值得一提的是：这种压缩是“无损压缩”，即不造成信息损耗的压缩，在需要时，

可采用特定的算法将图像复原成栅格形式。

本算法的主要特点有两个：

其一是编码转换。采用先构造“编码基值表”，然后，在遇到“1”值点时，再利用查表和相加的方法获得该点的编码。这样就使变换所需的计算量大大减少，也避免了用“字节内插”方法获得编码时由于图像大小的变化而带来的字节、位的数目不同在处理上的若干麻烦。

其二是建树过程。采用 2DRE 线性四叉树结构。另外，由于每次只输入图像的一行，建树时随时归并，这就大大地减少了冗余信息可能带来的“溢出”。因为在图像较大时，内存容量是否够用是一个普遍性的问题，所以本算法的这一特点，并不仅仅是一种权宜之计，而是提供了一种可能，即在内存相对较小而图像相对较大的情况下使变换得以进行。

运行本算法的结果表明，其时间复杂度与图像大小成比例关系。图像越大即像元总数越大，则所需运算时间越多。确切地说，时间复杂度与图像中“1”值点数量成比例关系。“1”值点越多，算法所需的插入、归并时间就越多。至于其空间复杂度，由于本算法对图像采取逐行读入、处理，逐层归并的方法，故所需的内存仅仅是可容纳一行图像和一棵四叉树，以及编码基值表等若干工作单元这样一些空间，所以空间效率是较高的。这样，就可以避免有时因图像较大而不能同时装入内存引起的矛盾和在冗余信息较多时先建树，最后归并的处理方法可能造成的“溢出”情况。

参　考　文　献

[1] A. Klinger & C. R. Dyer, Experiments on Picture Representation Using Regular Decomposition, Computer Graphics and Image Processing, Vol. 5, 1976.

[2] I. Gargantini, An Effective Way to Represent Quadtrees, Commu. ACM, Vol. 25, No. 12, 1982.

[3] J. P. Lauzon et al., Two-Dimensional Run-Encoding for Quadtree Representation, Computer Vision Graphics and Image Processing, Vol. 30, 1985.

[4] D. J. Abel, A Data Structure and Algorithm Based on a Linear Key for a Rectangle Retrieval Problem, Computer Vision Graphics and Image Processing, Vol. 24, 1983.

[5] D. J. Abel, A B -Tree Structure for Large Quadtrees, Computer Vision Graphics and Image Processing, Vol. 27, 1984.

[6] F. D. Libera & R. Gosen, Using B-Tree to Solve Geographic Range Queries, The Computer Journal, Vol. 29, No. 2, 1986.

[7] S. K. Bhaskar & A. Rosenfeld, Parallel Processing of Regions Represented by Linear Quadtrees, Computer Vision Graphics and Image Processing, Vol. 42, 1988.

[8] D. M. Mark, J. P. Lauzon & J. A. Cebrian, A Review of Quadtree-Based Strategies for Interfacing Coverage Data With Digital Elevation Model in Grid Form, Geographical Information System Vol. 3, No. 1, 1989.

THE ALGORITHM CONVERTING RASTER TO 2DRE QUADTREE STRUCTURE

Xiao Ke

(Institute of Remote Sensing Applications, Chinese Academy of Sciences)

Abstract

This paper advances and analyses an algorithm converting Raster to Quadtree. Both of the two data structures have their own advantages and disadvantages and suit different processes and operations for corresponding purposes. Raster is the most common structure for image data; and quadtree is a new data structure which has drawn more and more attention and has been developed very fast in recent years. The paper reviews the developing history of quadtrees, describes the converting algorithm and the base of it in detail. Furthermore, the algorithm is evaluated and analysed.

Key words Remote Sensing Image Processing Data structure Quadtree Geographic Information System