环 境 遥 感
第 8 卷 第 1 期 REMOTE SENSING OF ENVIRONMENT Vol. 8, No. 1
1993 年 2 月 CHINA Feb., 1993

2DRE 四叉树—栅格结构的变换算法

萧 柯
(中国科学院遥感应用研究所)
1991 年 12 月 11 日收稿

摘 要

本文提出并分析了 2DRE 四叉树到栅格结构的变换算法。这一算法可看作栅格—2DRE 四叉树变换算法的逆变换，但不是它的"反演"，而是采用了较为简洁的求交集运算以及栅格— 2DRE 四叉树变换算法中的编码转换方法，使得这一算法的实现更加有效，且避免了在图像较大而内存较小的情况下可能发生的"溢出"等矛盾。

关键词 遥感 地理信息系统 图像处理 数据结构 四叉树 栅格

从遥感的角度看，地理信息系统是现有的数据处理系统的延长和扩充。其核心是遥感图像处理系统、数据库管理系统以及在其之上的、能够迅速、准确地对图像数据进行录入、存储、检索、更新和一系列运算(包括对系统实体的空间和非空间属性的处理)的软件发展。在这中间，图像数据结构以及相应算法的研究具有不言而喻的重要意义。2DRE 四叉树[1]和栅格这两种结构各具特点，从而决定了它们在图像数据结构及算法研究，图像处理和地理信息系统中的地位。前者除了具有清晰、紧凑的优点之外，还适合于进行某些基本的处理运算；而后者除了作为图像的最初获取形式以外，现有的输出设备也多是以这种形式进行逐行输出的。一个完整的信息处理系统，应具有不同的数据结构，以满足不同的处理要求，且它们之间能够有效地相互变换。

在本文中，作者提出并分析了 2DRE 四叉树到栅格结构的变换算法，首先介绍该算法的预备知识；进而详述该算法；最后分析、讨论该算法的特点和时间、空间复杂度。

一、算法变换基础

1. 编码转换

设有一幅 $2^n \times 2^n$ 的数字图像，$[i, j]$ 表示图像中任意像元 P 的 X, Y 座标，令 $Q(i, j)$ 为 $x = i$, $y = j$ 处的 2DRE 四叉树编码，根据这种编码的构成规则，我们有：

$$Q(i,j) = Q(i,0) + Q(0,j) = Q(i,0) + 2Q(j,0) \qquad (1)$$

由此，只要得到图像中第一行 (y 座标 $j = 0$) 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。根据文献 [2] 中的 (8)—(11) 式，可得到 $Q(i, 0)$，再根据上面 (1) 式，可得到图像中全部点的编码值。这样就完成了从 X, Y 座标到 2DRE 四叉

树编码之间的转换。

2. 2DRE 四叉树的交集运算

将一幅图像的 2DRE 四叉树看作一个集合，其元素为编码连续的区域。由此，可将集合运算引入这一区域。两幅图像的交、并、差运算，可以用其各自的 2DRE 四叉树间的相应运算来实现[3]。

给定两棵 2DRE 四叉树 A 和 B，则 $A\cap B$，$A\cup B$，$A-B$ 和 $B-A$ 的结果仍是 2DRE 四叉树。这些运算实际上是一些更复杂的运算的基础。(这里仅讨论求交集的

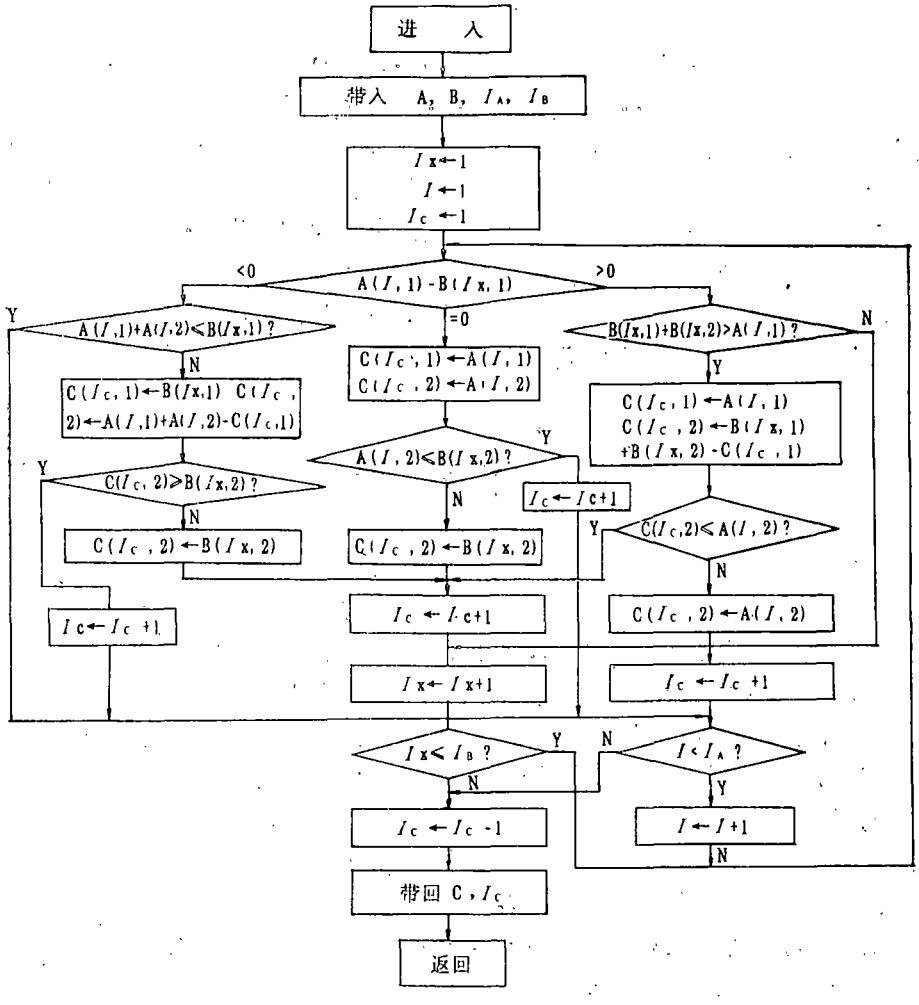

图 1　对集合 A 和 B 作求交集运算的算法程序框图

注：其中 A，B 为参加运算的四叉树集合文件；C 为 $A\cap B$ 的结果。I_A、I_B、I_C 分别为 A、B、C 三集合中的元素数目

Fig. 1　The Disgram of the Program for intersection operation between set A and B

Note: In fig. 1, A and B are the sets in operation; C is the result of the intersection between A and B; I_A, I_B, I_C are the element number in set A, B and C, respectively.

运算)。

给定一个代表图像 P_A 的 2DRE 四叉树的集合 A，设其中的编码连续区域有 I_A 个，于是可记作：

$$A = \{A_1, A_2, \cdots, A_{I_A}\}$$

其中 $A_i(i = 1, 2, \cdots, I_A)$ 是编码连续区域，i 是区域的编号。由 2DRE 四叉树的性质，可知集合 A 中的元素 A_i 是有序的，且 A_i 本身又是由一系列编码值构成的有序集。不难看出，求两个集合 A 和 B 的交集，实际上是求它们各自的子集之间的交集。设：

$$A = \{A_1, A_2, \cdots, A_{I_A}\} \tag{2}$$

$$B = \{B_1, B_2, \cdots, B_{I_B}\} \tag{3}$$

则：

$$A \cap B = \{A_1 \cap B_1, A_1 \cap B_2, \cdots, A_1 \cap B_{I_B}, A_2 \cap B_1, \cdots, A_2 \cap B_{I_B}, \cdots, A_{I_A} \cap B_{I_B}\} \tag{4}$$

看起来比较复杂，但是，由于这种集合的元素 A_i 和 B_j 各自的排列都是有序的，且本身都是由一系列连续的整数(即编码值)构成的另一类集合。又根据 2DRE 四叉树的性质和记录方式，A_i 和 B_j 中的元素可由它们的两个属性值来代表：$A_{i,1}$ 和 $B_{j,1}$ 分别代表它们的区域中的首编码；$A_{i,2}$ 和 $B_{i,2}$ 则分别代表它们的区域中的编码数目(即值点数目)。于是，我们就可以把这类集合运算转化为算术运算来进行，从而使问题的解决大大简化，并以此作为 2DRE 四叉树一栅格结构变换算法的核心。求交集运算的过程如图 1 所示。

二、算 法 描 述

2DRE Quadtree—Raster 变换算法对图像的 2DRE 四叉树文件进行变换，逐次得到图像的各行，并将其输出，从而使逐行输出的设备能够按顺序输出由 2DRE 四叉树文件给定的图像，而无需相应于图像的 $2^n \times 2^n$ 的矩阵存储空间。本算法可以看作是 Raster—2DRE Quadtree 变换算法[2]的逆变换，但却不是将该算法“反演”，而是采用了别的手段。即上一节中讨论的求交集运算。另外，还采用了该算法中的获取四叉树编码的方法。由于不需要庞大的矩阵存储空间，就使得本算法能够有效地利用内存，从而避免了“装不下”整幅图像的情况。

设所求图像为 $2^n \times 2^n$，本算法的处理过程如下：

(1) 读入 2DRE 四叉树集合文件 B。

(2) 根据 [2] 中 (8) 式和 (9) 式求出第一行中各元的编码值 $Q(i, 0)$ $(i = 0, 1, \cdots, 2^n - 1)$ 作为编码基值表。

(3) 置输出图像行计数器 j 的初值为 0。

(4) 根据行号 j 和文献 [2] 中 (7) 式，得到本行中各元的编码值 $Q(i, j)$ $(i = 0, 1, \cdots, 2^n - 1)$。

(5) 构成本行的四叉树集合 A_i 形如表 1。

(6) 对 A_i 和 B 进行求交集运算，得到 $C = A_i \cap B$。

表 1　第 j 行图像的 2DRE 四叉树集合 A_j

Table 1　The 2DRE Quadtree Set A_j of the jth Row of Image

$Q(0,j)$	$Q(1,j)$	…	$Q(2^n-1,j)$
1	1	…	1

(7) 将 C 与 A_j 逐元进行匹配，匹配点置 1；不匹配点置 0。得到第 j 行图像。

(8) 输出第 j 行图像。

(9) $j \leftarrow j+1$。

(10) 判断是否 $j < 2^n$。是则转到第 (4) 步。否则结束运算。

以下用一系列图表来进一步说明本算法。设所求图像为 $2^3 \times 2^3$ 者。给定其 2DRE 四叉树集合如表 1 所示。运算时先按行得到四叉树的完全编码，再逐行与该集合进行求交集的运算，由此即得到各行的输出图像。

表 2　有待于进行变换的 2DRE 四叉树文件 B

Table 2　2DRE Quadtree set B before Converting

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

表 3　第一行图像的 2DRE 四叉树 A_1

Table 3　The 2DRE Quadtree Set A_1 of the First Row of Image

0	1	4	5	16	17	20	21
1	1	1	1	1	1	1	1

表 4　$A_1 \cap B$ 的结果四叉树

Table 4　The Result Image of $A_1 \cap B$

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

表 5　第二行图像的 2DRE 四叉树 A_2

Table 5　The 2DRE Quadtree Set A_2 of the Second Row of Image

2	3	6	7	18	19	22	23
1	1	1	1	1	1	1	1

表 6　$A_2 \cap B$ 的结果四叉树

Table 6　The Result Image of $A_2 \cap B$

2	3	7	18	19	22
1	1	1	1	1	1

表 7 第八行图像的 2DRE 四叉树 A_8

Table 7 The 2DRE Quadtree Set A_8 of the eighth Row of Image

42	43	46	47	58	59	62	63
1	1	1	1	1	1	1	1

表 8 $A_8 \cap B$ 的结果四叉树

Table 8 The Result Image of $A_8 \cap B$

42	43	46	47	58
1	1	1	1	1

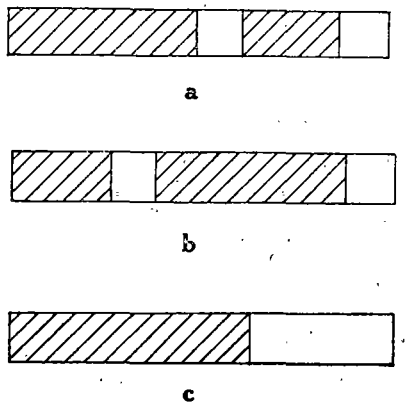

图 2 若干行结果图像

(a) $A_1 \cap B$ 得到的第一行图像，(b)$A_2 \cap B$ 得到的第二行图像，(c)$A_8 \cap B$ 得到的第 8 行图像

Fig. 2 Some Rows of the Result Image

(a) First row of result Image from $A_1 \cap B$

(b) Second row of result Image from $A_2 \cap B$

(c) Eighth row of result image from $A_8 \cap B$

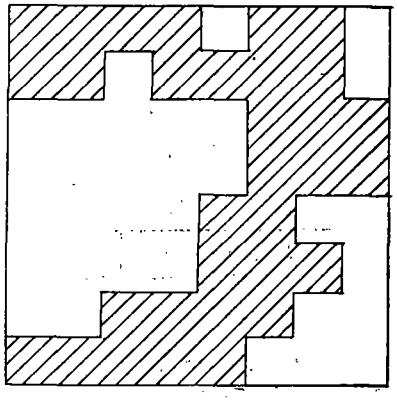

图 3 经四叉树—栅格变换所得到的完整图像（其 2DRE 四叉树如表 2 所示）

Fig.3 The Whole Result Image after the Conversion of 2DRE Quadtree to Raster (Its 2DRE quadtree is showed in Table 2)

表 2 是有待于进行变换的 2DRE 四叉树文件 B；表 3、表 5、表 7 分别是图像的第一、二、八行的 2DRE 四叉树集合 A_1，A_2 和 A_8；表 4、表 6、表 8 则分别是 A_1，A_2，A_8 与 B 求交集的结果四叉树；图 2 示出了由它们得到的第一、二、八行图像；图 3 则是经过变换的相对于集合 B 的完整图像。

三、算法的分析与评价

2DRE 四叉树到栅格结构的变换算法之主要特点是：

第一，节省空间。本算法由于采用的是得到一行图像即输出，而不在内存中保留整幅图像的“流水”处理方法，就避免了因所要求的内存空间过大而使处理过程无法进行的情况，也可以避免处理过程中间可能发生的“溢出”。这样，就使本算法能够处理较大的图

像。事实上，本算法所需的工作空间只限于存放编码基值表，存放 2DRE 四叉树文件，每一行的编码以及一行图像所需要的缓冲区这几项。所以，这方面的特点是较为突出的。

第二，过程简洁。由于采用的是求交集运算方法来获得每行图像，图像的每一行，行中每一像元的顺序均隐含规定。这就避免了已有的某些算法采取的、先对整个四叉树文件进行编码到座标的转换，再对座标值进行排序，最后再得到图像的冗长过程。同时，也避免了上述处理方法对于空间、时间的过高要求以及可能带来的一些困扰。

关于本算法的时间、空间复杂度问题。从前述的特点中可以看出，它的空间效率是较高的。所需的工作空间仅是一棵 2DRE 四叉树(二维数组形式) 和一行图像的输出缓冲区以及不多的诸如编码基值表和一行图像的完全编码等辅助空间。至于其运算时间，由于每得到一行图像，需调用一次求交集的算法；又因为在本算法中求交集是以像元为单位逐一进行，所以，不难看出，计算时间是与图像的值点数目成比例关系的。另外，由于每得到一行图像即需进行输出，则 I/O 时间与图像的行数成比例关系。 在实际处理过程中，如果存储空间允许，可在得到若干行图像后再输出，以减少输出次数，节省 I/O 时间，同时也就缩短了整个运算的时间。

在本算法的描述中，是通过对二值图像进行处理来说明的。在实际应用中，可逐一用不同灰阶的四叉树与各行图像的四叉树子集进行求交集运算，在第七步“匹配”时，对相应灰阶的值点赋给该灰阶值，即可得到具有全部灰阶的输出图像。

参　考　文　献

[1] J. P. Lanzon et al., Two-Dimensional Run-Encoding for Quadtree Representation, Computer Vision Graphics and Image Processing, Vol. 30, 1985.
[2] 萧柯，遥感图像的栅格— 2DRE 四叉树变换算法，环境遥感，6(4)，1991。
[3] 萧柯，2DRE 四叉树的集合运算，环境遥感 7(4)，1992。

THE ALGORITHM CONVERTION 2DRE QUADTREE TO RASTER

Xiao Ke

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

Abstract

This paper advances and analyses an algorithm converting 2DRE quadtree to raster. The algorithm can be served as the inverse operation of raster to 2DRE quadtree, but it is not direct inversion, instead, it takes other simpler measur i.e. intersection of 2DRE quadtree and code transforming method in Raster to Quadtree conversion, to make the algorithm more effective. It can also avoid possible trouble of memory overflow caused by the operation on bigger images in smaller memory.

Key words Remote sensing Geographic infomration system Image processing data structure Quadtree Raster