环 境 遥 感
第7卷 第3期 REMOTE SENSING OF ENVIRONMENT Vol. 7, No. 3
1992 年 8 月 CHINA Aug., 1992

图像 2DRE 四叉树—链式码向量的变换算法

萧 柯
(中国科学院遥感应用研究所)

1991 年 4 月 4 日收稿

摘 要

本文叙述了图像数据结构的一种变换算法。近年发展起来的四叉树数据结构,是描述图像结构、压缩存储空间等方面的一种好方法;而链式码向量在检测和描述图像区域边界及其形状方面有很多长处。但两种结构各有其缺点,它们互相补充,才能得到较好效果。本文提出并分析了一个从 2DRE 四叉树得到图像中的区域边界的算法,即 2DRE 四叉树—链式码向量变换算法(简称 Q—V 算法)。文中介绍了算法的基础和算法本身,并对其特点和效果进行评价和分析。

关键词 图像处理 数据结构 四叉树 向量 区域边界

在遥感图像处理和地理信息系统的实际工作中,不同数据结构间的变换算法是一个重要的研究课题。四叉树,特别是 2DRE 四叉树是一种决定于图像自身性质,且具有清晰、简单、规整,便于检索等特点的数据结构;而链式码向量结构尽管在某些区域运算上不及四叉树和栅格等结构方便,但它作为区域边界的一种表示方法,在边界检测,如检测拐点、凸凹等情形的功能上以及在表示区域形状时,其作用却是其它结构所不能比拟的。在很多情况下,人们不仅关心区域的结构,也关心区域的边界、形状,而链式码向量结构无疑是一种适宜的表示方式。这样,研究四叉树—链式码向量即 Q—V 的变换算法,在实际工作中的意义就不言而喻了。

本文先介绍算法的基础,即链式码向量结构以及区域边界的跟踪问题,然后描述算法并分析其应用效果。

一、区域边界与链式码

给出图像中的区域,其边界可看作平面上的图形,或封闭曲线,据此,可以找到一种既简单,又清晰的表示方法。

如图 1 所示,用整数 $i=0,1,2,3$ 分别表示四个方向,即顺时针旋转 $90°\times i$ 的方向。这样,就可以用一系列不同方向连接起来的单位向量来表示平面上的曲线,从而作为封闭曲线的区域边界也就可以用这种方法来表示了,这种向量序列,称之为链式码[1]。

图 2 表示图像中的区域,表 1 为图 2 中区域边界自 A 点开始的链式码。在链式码序列中,封闭曲线(即区域边界)上的拐点或凸凹、平直等特征,用数码表示十分清楚,可以根据链式码序列,勾画出区域边界的形状。

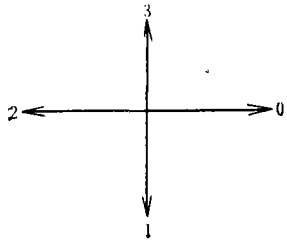

图 1　方向码的规定

Fig. 1　The rule of direction codes

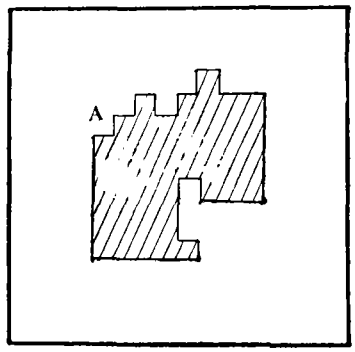

图 2　图像上的区域

Fig. 2　A field in an image

表 1　区域边界的链式码

Table 1　The chain code of the field's edge

0	3	0	1	0	3	0	3	0	1
0	0	1	1	1	1	1	2	2	2
3	2	1	1	1	0	1	2	2	2
2	2	3	3	3	3	3	3	0	3

二、边 界 跟 踪

设有一棵表示某一图像的 2DRE 四叉树。其元素即图像中的 1 值点连续区，而所谓“区域”正是由若干个这样的连续区构成。

从上一节的讨论可知，对图像中的区域，只要任意给定其边界上的某一点，就可以从该点出发，根据一系列单位向量的方向，确定其边界曲线的链式码。由于曲线是封闭的，最后一个单位向量必然指向起点。这样，给定了起点和一系列链式码，区域边界的形状和位置也就完全确定了。因为曲线是封闭的，所以边界上的任一点都可以作为曲线的起点，为方便起见，把四叉树中的第一个元作为边界曲线的起点。其原因：第一，它在区域边界上；第二，它位于所在区域的左上角（但不一定最左或最上，正如同数学中光滑曲线上的极大值点不一定就是最大值点的一样道理），从而便于确定初始的跟踪方向。

在开始跟踪之前，首先确定初始的边界点对 $(P_{1,1}, P_{1,0})$。其中 $P_{1,1}$ 是值点，即区域中的点；$P_{1,0}$ 则是背景点。前面已提到，边界的起点规定为四叉树中首元代表的点。该点即作为初始点对中的 $P_{1,1}$。又因 $P_{1,1}$ 处在区域的左上角，故规定初始方向为 0。于是，$P_{1,0}$ 与 $P_{1,1}$ 的关系为：

$$X_{P_{1,0}} = X_{P_{1,1}} \tag{1}$$

$$Y_{P_{1,0}} = Y_{P_{1,1}} - 1 \tag{2}$$

有了初始点对 $(P_{1,1}, P_{1,0})$ 和初始方向 0，就可以寻找边界的下一个点对和方向，这

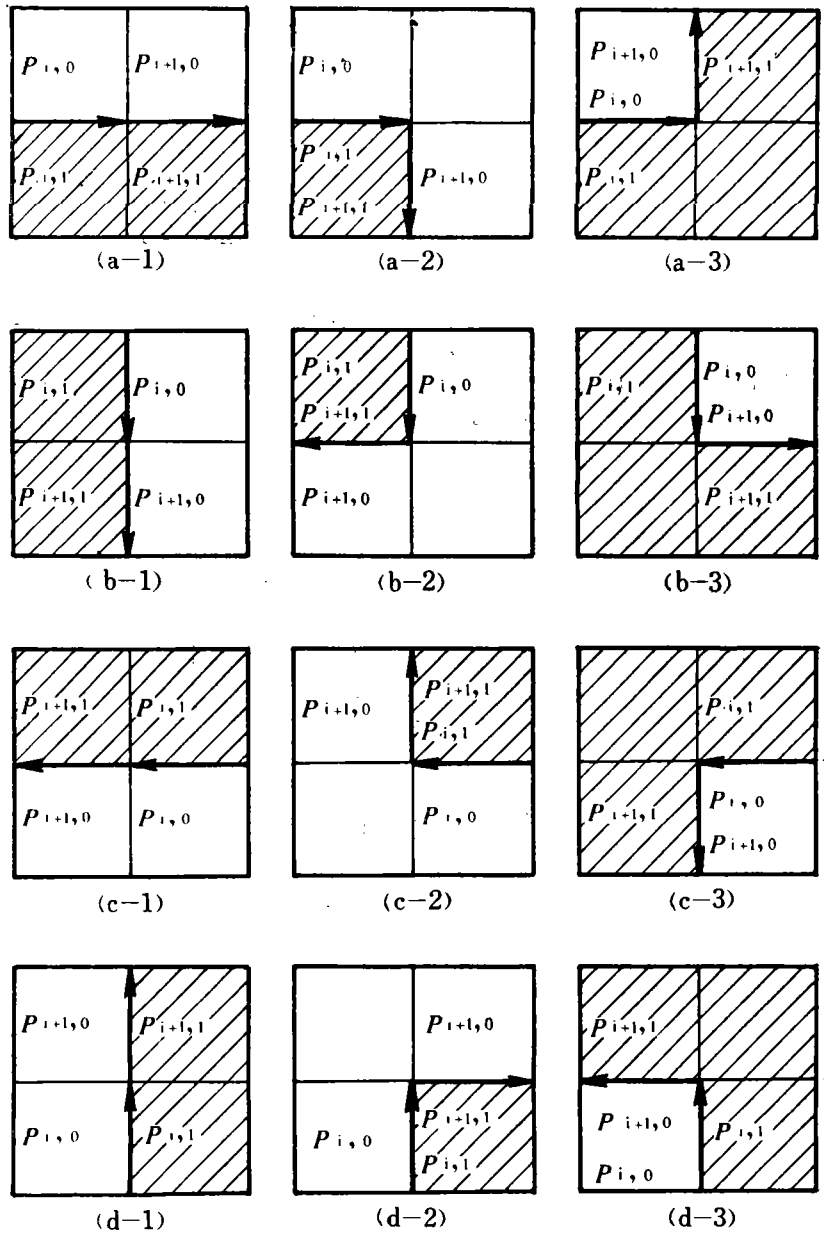

图 3 边界跟踪的几种可能情况

(a-1)	$D_i = 0$	$D_{i+1} = 0$	(b-1)	$D_i = 1$	$D_{i+1} = 1$
(a-2)	$D_i = 0$	$D_{i+1} = 1$	(b-2)	$D_i = 1$	$D_{i+1} = 2$
(a-3)	$D_i = 0$	$D_{i+1} = 3$	(b-3)	$D_i = 1$	$D_{i+1} = 0$
(c-1)	$D_i = 2$	$D_{i+1} = 2$	(d-1)	$D_i = 3$	$D_{i+1} = 3$
(c-2)	$D_i = 2$	$D_{i+1} = 3$	(d-2)	$D_i = 3$	$D_{i+1} = 0$
(c-3)	$D_i = 2$	$D_{i+1} = 1$	(d-3)	$D_i = 3$	$D_{i+1} = 2$

Fig. 3 Possible Situations of edge tracking

样一个个找下去，就实现了对整个边界的跟踪。由于我们规定构成边界曲线的单位向量具有四个方向，所以，已知一点对（$P_{i,1}$，$P_{i,0}$）和方向 D_i，欲求（$P_{i+1,1}$，$P_{i+1,0}$）和 D_{i+1}，不外乎图 3 所示的几种情况(按顺时针方向行进，即区域总在行进方向之右)。

从图 3 可知，不论方向码为 0，1，2 或是 3，都只有三种可能的行进方向，即向前，向左，向右。用公式表示这三种情况：

$$\text{向前：}\ D_{i+1} = D_i \tag{3}$$

$$\text{向右：}\ D_{i+1} = (D_i + 1)_{\bmod 4} \tag{4}$$

$$\text{向左：}\ D_{i+1} = (D_i + 3)_{\bmod 4} \tag{5}$$

对应不同的方向码，点对的坐标之间有以下关系：

$D_i = 0$ 时，如式(1)，式(2)所示

$D_i = 1$ 时，

$$X_{P_{i,0}} = X_{P_{i,1}} + 1$$

$$Y_{P_{i,0}} = Y_{P_{i,1}} \tag{6}$$

$D_i = 2$ 时

$$X_{P_{i,0}} = X_{P_{i,1}}$$

$$Y_{P_{i,0}} = Y_{P_{i,1}} + 1 \tag{7}$$

$D_i = 3$ 时

$$X_{P_{i,0}} = X_{P_{i,1}} - 1$$

$$Y_{P_{i,0}} = Y_{P_{i,1}} \tag{8}$$

这样，根据(1—8)式所叙的规则，就可以对边界进行跟踪，从而求得整条边界曲线。在跟踪过程中，要随时对点对（$P_{i,1}$，$P_{i,0}$）进行检测，以确定曲线的走向，同时也检测是否已回到曲线的起点，以确定是否结束本曲线的跟踪。由于欲求其边界的区域是由 2DRE 线性四叉树所给定，跟踪过程中的检测和下一步的点对的确定，都需要进行“编码一坐标”或“坐标一编码”转换[2]。前者用于获得点对，以便记录；后者则得到编码，用以检索四叉树文件，来确定所求得的点究竟是区域点还是背景点。这种检索还有一个用途，当图像的区域不止一个时，就要进行多于一次的边界跟踪过程。根据 2DRE 线性四叉树的构造，可知在这种情况下，如果边界跟踪的次数少于图像的实际区域数时，四叉树中必然有某些集合元素即编码连续区未被访问到。于是，我们可以通过这种检索将已被访问到的区号“留迹”。在每一次跟踪过程完毕，即检查区号留迹表。如没有未被访问到的区，则整个跟踪过程结束；反之，则从序号最小的未访问区的首元开始下一轮次的边界跟踪，以确定下一区域的边界，这样就避免了可能发生的遗漏。

根据以上原则和方法，就解决了 Q—V 算法的核心——边界跟踪问题。

三、算法的描述

Q—V 变换算法的处理对象是图像的 2DRE 线性四叉树文件，用于从四叉树所表示的图像中获得其中区域的边界，采用记录边界曲线上各点的坐标的方式来标识所获得的边界点序列，即用坐标形式记录链式码向量结构。这样处理便于将位于同一直线(0°，45°，90°，135°)上的点归并，仅记录直线段的首尾坐标，以节省空间，而且便于以后的图形输

表 2 图像四叉树文件 B

Table 2 The 2DRE Quadtree File B

连续区号	①	②	③	④	⑤	⑥	⑦	⑧
编 码	3	9	11	17	19	26	36	48
	1	1	5	1	5	1	4	16

出。

设给定图像的 2DRE 四叉树文件 B (见表 2, 图 4), 图 4 为表 2 的实际图像。以下结合图 4 实际图像简述算法获得区域边界曲线的过程:

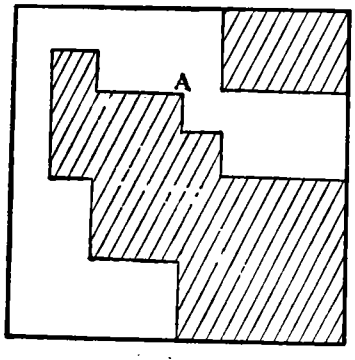

图 4 文件 B 的实际图像

Fig. 4 The Image from which File B is Obtained

1. 根据文献[2]中式(3)—式(10)求出编码基值表。

2. 置方向码 $ISB \leftarrow 0$; 置初始区号 $K \leftarrow 1$。

3. 置向量区 $IBOD$ 的指针 $NP \leftarrow 0$。

4. 将 B(K, 1) 作为边界起点 AIN, 并置 $Q_1 \leftarrow$ AIN。

5. 根据文献[2]中式(1)、式(2)和式(11)—式(14)求得 AIN 的坐标 X_1, Y_1。

6. 由 X_1, Y_1 和方向码 ISB 求得 X_0, Y_0。

7. 根据 X_0, Y_0 和编码基值得到 Q_0, 从而得到初始点对的全部信息 ($P_{1,1}$, $P_{1,0}$) 各自的编码和坐标, 见表 3。

8. $IBOD(NP, 1) \leftarrow X_1$; $IBOD(NP, 2) \leftarrow Y_1$; 再 $NP \leftarrow NP + 1$。

9. 将区号留迹表的第 K 位置成 1。

10. 根据方向码和边界点对进行边界跟踪。先求出新的点对, 然后验证其二点是否各自合法, 如合法, 记录 X_1, Y_1, 检索 Q_1 属 B 中第几区, 记录其区号。然后得到新方向码 ISB 和当前点对信息; 如不合法, 则调整方向码 ISB, 再调整点对信息。(同线归并前的链码区的例子, 见表 4。)

表 3 初始点对信息

Table 3 Initial point pair information

Q_1	3	Q_0	1
X_1	1	X_0	1
Y_1	1	Y_0	0

注: 由算法 4—7 步得到。

表 4 同线归并前的链式码

Table 4 Chain code area before same line merging

X	1	1	2	3
Y	1	2	2	2

11. 判定是否 NP 大于 4，是则判别最近的三个 $IBOD$ 中记录的点是否需要归并，如是则进行归并(表 4、表 5 的例子为 A 点处同线归并前后的链码区；表 6 为跟踪至 A 点处时的区号留迹表)。然后调整链码区内容，$NP \leftarrow NP - 1$；否则往下进行。

12. 判别当前 Q_1 是否等于 AIN，是，则本次跟踪完毕，在链码区中给出标志(连续 5 个 0 如表 7 所示)，再转入下一步，否则转到第 10 步。

第一轮边界跟踪完毕的某些状态信息，其链码区的内容见表 7。

表 5　同线归并后的链码区

Table 5　Chain code area after same line Merging

1	1	3
1	2	2

表 6　区号留迹表

Table 6　Field number recording table

1	1	1	0	0	0	0	0

表 7　链码区的内容

Table 7　The content of Chain Code Area

1	1	3	3	4	4	7	7	4	4	2	2	1	1	0	0	0	0	0
1	2	2	3	3	4	4	7	7	5	5	3	3	1	0	0	0	0	0

13. 检索区域留迹表，带回标志信息 IV 和当前最小的未访问区号 KP (第一轮边界跟踪中，经若干次第 9 步操作，目前最小未访问区号=4，见表 8)。

表 8　区号留迹表内容

Table 8　The content of field number recording table

1	1	1	0	0	1	1	1

14. 判别 IV 是否为 0，是，则表示所有区均已遍访完成，转向第 17 步，否则转下一步。

15. 逐一检验 B 中 第KF 区中的各元是否具有边界点的性质(边界点至少有一邻点是背景点，即不在 B 中)。如找到则将该点的编码作为 Q_1，转下一步；如遍访本区无此类点，则将 KF 记入区号留迹表，再转第 13 步。

16. AIN $\leftarrow Q_1$；转第 5 步。(经 5—7 步操作得到第二轮边界跟踪前的初始点对信息

表 9　第二轮边界跟踪前的初始点对信息

Table 9　Initial point pair information of second round tracking

Q_1	17	Q_0	−1
X_1	5	X_0	5
Y_1	0	Y_0	−1

注：在本算法中，凡图像以外的点(即 X 或 $Y > 2^n - 1$；X 或 $Y < 0$) 其编码规定为 −1。

表 10 链码区的内容

Table 10 The content of chain code Area

1	1	3	3	4	4	7	7	4	4	2	2	1	1	0	0	0	0	0	5	7	7	5	5	0	0	0	0	0
1	2	2	3	3	4	4	7	7	5	5	3	3	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

表 11 区号留迹表内容

Table 11 The content of field number recording

1	1	1	1	1	1	1	1

的例子，见表 9。）

17. 结束本算法，输出所得链码（表 10、表 11 为实例中全部过程结束后的状态信息）。

四、Q—V 变换算法评价

根据 Q—V 变换算法，经过一系列的跟踪过程，求得图 4 所示图像（由表 2 给出的四叉树表示）中区域边界曲线的链码串的坐标形式如表 10 所示。由于采用了随时进行同线归并的策略，减少了大量的冗余信息，使得到的链码串尽可能紧凑，当然也就提高了空间的利用率。

本算法的主要特点是：

第一，使用方便。它可以直接用于图像的 2DRE 线性四叉树形式，而无需先将四叉树变换为栅格图像，再进行栅格到矢量变换的繁琐过程。特别是在需要对图像进行诸如并、交、补、差等集合运算，又希望看到每一步运算后图像中的区域形状时，直接运用本算法进行 Q—V 变换，要比每一步都先进行四叉树到栅格，再进行栅格到矢量变换要简单、方便得多。

第二，节省时间、空间。由于本算法是从四叉树直接得到区域边界的链式码串，从而所需空间仅是四叉树文件和链码区，而不需要变换到栅格结构时所必需的庞大的矩阵空间。在实际图像很大的情况下，这一特点更加突出。因为如果在常规情况下，先进行四叉树到栅格变换，再进行栅格到矢量变换，机器的内存就可能容纳不下原始的栅格图像，从而带来诸多的困难。而使用本算法则有效地克服了上述矛盾。另外，由于本算法实行的是边界跟踪，所处理的像元仅是区域边界上的值点和背景点，而不需象传统的栅格—矢量算法那样，对整幅图像的所有像元均加以判别，这就大大节省了处理时间。特别在图像较大，而其中区域的周长与面积的比值较小时，其效果更加明显。

在具体的程序编制上，也采取了若干措施，以进一步节省时间和空间，如果采用“同线归并”即仅记录直或斜线的首尾坐标而不是逐点记录，对区号留迹表用位，而不是用字节来记录信息，使这部分空间压缩至最小。检索四叉树文件时，根据其中连续区的数目多少，决定采取“对半检索”或“顺序检索”的策略，以节省这部分计算所需的时间。

关于本算法的时间、空间复杂度问题，从以上讨论中可以看出，其所需空间仅为四叉

树文件、链码区以及编码转换基值表所需要的，又由于本算法只作用于图像中区域的边界，即仅仅边界上的值点及背景点参加运算，所以本算法的运算时间与图像中各区域的周长之和成比例关系，区域越多周长越大，则运算时间越多，反之则越少。

参　考　文　献

[1] A. Rosenfeld, A. C. Kak, Digitel Picture Processing, Academic Press, 1982.
[2] 萧　柯，遥感图像栅格—2DRE 四叉树结构变换算法，环境遥感，6(4)，1991。

THE CONVERTING ALGORITHM OF 2DRE QUADTREE-TO-VECTOR

Xiao Ke

(Institute of Remote Sensing Application Chinese Academy of Sciences)

Abstract

Quadtrees, the data structure developed in recent years, have advantages in expressing structure and compressing the space of storage while Chaincode-Vector are the data structure used to detect and describe the edges and shapes of fields in an image, so both of them have their own advantages and disadvantages in different situations. This paper advances and analyses an algorithm of converting 2DRE Quadtree to Vector, which abstracts the edges of fields in an image from its 2DRE Quadtree. The fields' edges are represented in the form of coordinates of vectors for displaying the fields' shape and outputting the vectors easily. In this paper, the base of the algorithm is introduced first; then the algorithm is described in detail; finally the features and practical results of the algorithm are estimated and analysed.

Key words Image processing　Data structure　Quadtree　Vector　Field's edge