

WINGIS 面状图栅格数据向 矢量数据转换方法研究

李应国 高显连 田永林

摘要 通过对栅格数据结构的分析,采用在栅格点间插入边界点,将栅格数据转换成边界点栅格数据,边界点的取值非0即1。且其最大的方向数为4;根据边界点的方向数,设置搜索成线过程的起点和终点,搜索成线过程沿着方向数为2的边界点方向进行;完成搜索成线后,对线进行多余点处理及圆滑处理;然后进行多边形及其标志点的生成,建立面状图的拓扑关系并回填属性值,从而完成了包含拓扑关系及其属性的栅格向矢量数据转换。

关键词 GIS(地理信息系统)、栅格数据、矢量数据

地理信息系统(GIS)是利用计算机进行空间数据及属性数据管理的信息系统,广泛应用于与地理信息有关的国民经济的各个部门(如农、林、牧、水保、城市规划等),是进行资源管理、规划设计、数据更新等现代化的手段之一。WINGIS是一通用地理信息系统,该系统充分利用计算机资源,采用面向对象的程序设计技术(OOP),基于国际上流行的WINDOWS环境,具有界面友好、运行速度快、操作简单等特点。本文介绍WINGIS系统中栅格数据向矢量数据转换技术。

1 栅格数据与矢量数据

栅格数据与矢量数据是地理信息系统中采用的两种基本数据类型,空间数据的输入通常采用其中的一种,但在系统整个处理过程中,两者均经常用到。如在建立数字地形模型时,通常将矢量的地形数据转换为栅格数据,而用栅格地形数据形成的坡度、坡向数据,又经常需要转换为矢量数据。两者间的相互转换有时是必需的,有时是为简化或加快数据处理过程,因而栅格与矢量数据相互转换技术是地理信息系统所必备的能力之一。就技术方法而言,矢量数据转换为栅格数据已有很多成熟的算法,本文不再讨论。

栅格数据是以规则的阵列来表示所研究对象的一种数据类型,常用来表示面状图。在这种数据类型中,点(象元)的位置是由其所在阵列的行和列隐含表示的,点的属性则由点所在位置的象元取值来表示,面状对象的位置由一片相邻的象元的位置集合表示,面上的每一点至少有两个相邻的点,面的属性则由面上点的取值来表示,由于象元是有一定大小的,因而栅格数据所表示的面状对象是不连续的、其边缘是锯齿形状的,同一对象的全部象元的存储位置并不总在一起,因而无法对单个对象进行处理;矢量数据则是直接记录所研究对象的坐标和其属性值的一数据类型,常用来表示点、线或面状对象,在矢量数据类型中,点、线和面的属性由其属性

1994-07-05 收稿。

李应国助理研究员,高显连,田永林(中国林业科学研究院资源信息研究所 北京 100091)。

值表示,点的位置是由其坐标直接表示的,而线上一系列的点(这些点的位置通常是按一定顺序排列的)的坐标表示了线的空间位置,面的空间位置则由围成面的封闭边界的全部线来表示。另外,在矢量数据类型中,点是没有大小的,因而这种数据类型所表示的对象是圆滑的。两种数据类型对点、线及面状对象的表示方法见图 1。

由两种数据类型的比较可知,面状对象的栅格数据向矢量数据转换应满足 3 个条件:A:空间位置关系的转换,即将面状对象边缘象元的行列号值转换为相应的点的坐标值,并建立点之间的邻接关系,从而形成线;B:拓扑关系的形成,即确定面(多边形)是由哪些线闭合而成的;C:属性关系,即标定矢量数据表示的面的属性。

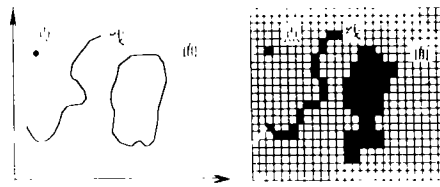


图 1 矢量数据(前)及栅格数据(后)对点、线及面状对象的表示方法

2 边界点数据的形成

假设有一 M 行 N 列的栅格数据 $D(M, N)$, 在 D 中有彼此相邻的 9 个栅格点 $A \sim I$ (图

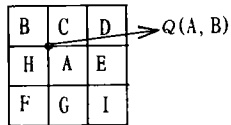


图 2 栅格数据 D

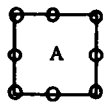


图 3 每个象元周围的八个边界点

2), 每个栅格点的属性值分别为 $V(A) \sim V(I)$, 两相邻栅格点的属性值之差为 S , 假定 $S=0$ (有时 $S \neq 0$), 当 $V(A) - V(B) = S$ 时, 栅格点 A 和 B 属于同一研究对象(线或面); 反之, 当 $V(A) - V(B) \neq S$ 时, 栅格点 A 和 B 不属于同一研究对象(线或面), 那么栅格点 A 和 B (有一定大小) 之间必有一边界点 $Q(A, B)$ (该点没有大小) 把栅格点 A 和 B 分开, 假定 A 的属性值 $V(A)$ 与它周围的其它 8 个栅格点的属性值之差都不等于 S , 栅格点 A 周围必有 8 个边界点(图 3), 由于栅格点 A 有一定的长(L)和宽(W), 因而栅格点 A 可以看作是由 8 个边界点围成的多边形; 在多边形的边界上, 两相邻边界点之间的距离等于 $L/2$, 或 $W/2$; 这样, 栅格数据 $D(M, N)$ 的全部栅格点的最多的边界点数为 $(2M+1)(2N+1) - MN$, 为讨论方便, 假定最外缘的边界点仍有 8 个边界点, 这些边界点形成了另外一栅格阵列 $U(2M+1, 2N+1)$ (图 4), 在栅格数据 U 中, 行列号同为单数(起始行列号都为 0)的栅格点不是边界点。因此上述过程可看作这样一数据变换, 即 $D(M, N) \Rightarrow U(2M+1, 2N+1)$ 。

事实上, 在栅格数据 $U(2M+1, 2N+1)$ 中, 上述的边界点并非全部是真正的边界点, 如图 2 中, 当 $V(A) - V(B) = S$ 时, 栅格点 A 和 B 属于同一研究对象(线或面), 栅格点 A 和 B 之间没有边界点 $Q(A, B)$, 为了区分一边界点是否真正的边界点, 可以给该边界点取不同的属性值, 即当一边界点是真正的边界点(下面提到的边界点都为真正的边界点)时, 令其属性值为 1, 否则为 0, 这样栅格数据 $U(2M+1, 2N+1)$ 的属性值只为 1 或 0, 在实际实现时, 可用一个位来表示一个栅格点的属性值, 栅格数据 U 共占存储空间 $[(2M+1)(2N+1)/8]$ 个字节。因而

3 边界点的取值与方向数

事实上, 在栅格数据 $U(2M+1, 2N+1)$ 中, 上述的边界点并非全部是真正的边界点, 如图 2 中, 当 $V(A) - V(B) = S$ 时, 栅格点 A 和 B 属于同一研究对象(线或面), 栅格点 A 和 B 之间没有边界点 $Q(A, B)$, 为了区分一边界点是否真正的边界点, 可以给该边界点取不同的属性值, 即当一边界点是真正的边界点(下面提到的边界点都为真正的边界点)时, 令其属性值为 1, 否则为 0, 这样栅格数据 $U(2M+1, 2N+1)$ 的属性值只为 1 或 0, 在实际实现时, 可用一个位来表示一个栅格点的属性值, 栅格数据 U 共占存储空间 $[(2M+1)(2N+1)/8]$ 个字节。因而

尽管 $D(M, N) \Rightarrow U(2M+1, 2N+1)$ 的过程中, 数据量增加了 4 倍 $[(2M+1)(2N+1)/MN]$ 多, 实际所需要的存储空间要小得多, 假如栅格数据 $D(M, N)$ 的属性值用一个字节表示, D 共占存储空间 MN 个字节, 即 $(M, N) \Rightarrow U(2M+1, 2N+1)$ 的过程中, 数据量压缩了近 50%。由于在搜索成线时, 只处理边界点数据, 总的处理数据量减少了 50%。

在栅格数据 $U(2M+1, 2N+1)$ 中, 一边界点必和它的边界点相连同, 和一边界点相连同的其它边界点的总数称为该边界点的方向数, 当 U 没有经过栅格矢量数据转换时, 边界点的方向数的最大值为 4, 最小值为 2(图 5), 点 A 的方向数为 4, 点 E 的方向数为 3, 点 C 的方向数为 2, 各点方向数均不为 1, 说明由全部相邻边界点围成的多边形都是封闭的。

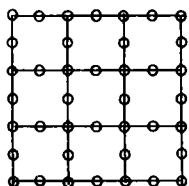


图 4 边界点阵列

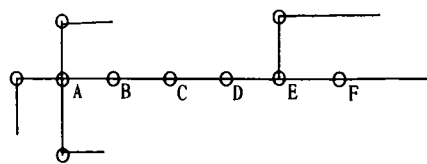


图 5 边界点的方向数

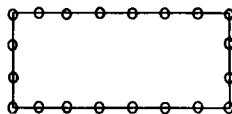


图 6 边界点构成岛

4 边界线的形成

将边界点按顺序连接便形成线, 将线上每点的行列值(或转换为相应的坐标值)以矢量数据记录下来, 便形成一条线的矢量数据, 这一过程是通过搜索边界点来完成的。

4.1 线的起始点和终点的确定

一边界点能否作为线的起始点(或终点)是由其方向数决定的, 在搜索成线的过程中, 应保持线的完整性, 不能使一条线断开。由图 5 可知, 当一边界点的方向数为 3 或 4 时, 该点必为多条线的交点(即结点), 如图 5 中的点 A(方向数为 4)或点 E(方向数为 3), 因而该点必为线的起始点(或终点); 方向数为 2 的边界点一般不能作为线的起始点(或终点), 如图 5 中的 C(或 B, D)点, 因为它将使一完整的线 ABCDE 断为两条(ABC 和 CDE), 但当边界点围成一封闭的环线(即岛)时(图 6), 环线上的每一边界点的方向数都为 2, 此时, 每边界点都可以作为线的起始点, 但不能作为线的终点; 由于在搜索过程中, 边界点的方向数将不断减少(后面将会看到), 因此在搜索过程的最后, 必然出现方向数为 1 的边界点, 此时这类边界点也作为线的起始点(或终点)。为了提高搜索成线的效率和速度, 应首先对方向数为 4 或 3 的边界点作为起始点进行搜索成线; 接下来以方向数为 1 的边界点作为起始点进行搜索; 最后以方向数为 2 的边界点作为起始点进行搜索, 当遇到方向数不为 2 的边界点或遇到和起始点相同的边界点时, 便结束该线。

4.2 线的搜索方向

当线的起始点确定后, 应首先循环判断和起始点相连同的其它边界点的方向数, 当遇到方向数为 2 的边界点(最少有一个这样的边界点)时, 搜索点就从起始点移至该点, 然后再从该点周围找出方向数为 2 且不是上一搜索点的边界点, 如此下去, 直至遇到线的终点。在搜索成线的过程中, 方向数为 2 的边界点(如图 5 中 B, C, D)只可能在一条线上, 当搜索过程经过该点后, 该点将不再是边界点, 即它的属性值将由 1 变为 0, 和该点相连同的其它边界点的方向数

将减少 1, 因此, 随着搜索的进行, 边界点数越来越少, 边界点的方向数将不断减少。

5 边界线的处理

搜索成线是逐个栅格点进行的, 由此形成的以矢量数据表示的线有两个问题: (1) 有许多点在同一直线上, 如图 5 中的线 ABCDE, 因而应进行多余点的删除; (2) 如果线有折点, 其必然成直角形状, 这样的线形同样造成了数据冗余, 且视觉效果很不理想, 必须进行圆滑处理。

5.1 多余点的删除

多余点的删除是基于这一思想: 设有连续的 3 点 $A(x_a, y_a)$, $B(x_b, y_b)$, $C(x_c, y_c)$, 当中间的 B 点位于由 A 点和 C 点所构成的直线上时, 即当 $(y_a - y_b)/(x_a - x_b) = (y_b - y_c)/(x_b - x_c)$ 时, B 点应删除, 否则 B 点保留。为防止分母为零的错误, 上式应写为: $(y_a - y_b)(x_b - x_c) = (x_a - x_b)(y_b - y_c)$ 。

5.2 线的圆滑处理

如图 7, DAE 中的点 A 为一直角折点, DA, EA 分别为一栅格点的长和宽, 在 $1/2DA$ 处加进一点 B, 在 $1/2EA$ 处加进一点 C, 然后删除 A 点, 此时线 DAE 变为线 DBCE, 直角折点 DAE 变为两个 135° 的折点。对全部直角折点进行上述处理后, 便形成一系列的 135° 的折点, 此时线形仍不够圆滑。因而在 $1/2DB$ 处, 加进一点 H, 在 $1/2BC$ 处加进一点 I, 在 $1/2CE$ 处加进一点 K, 然后删除 B, C 点, 此时线 DBCE 变为线 DHKE, 折点角度变得更大, 比较圆滑自然。为保证面状图拓扑关系不受破坏, 圆滑处理应保证起点及终点的位置不能有任何变动, 同时为进一步减小冗余度, 应再次进行多余点的删除处理。由于边界点数据每个栅格点的长和宽只是原始数据栅格点的长和宽的一半, 因此线的圆滑处理造成的误差不会超过一个象元。至此已将栅格数据面的边界线转换成了以矢量数据表示的线。

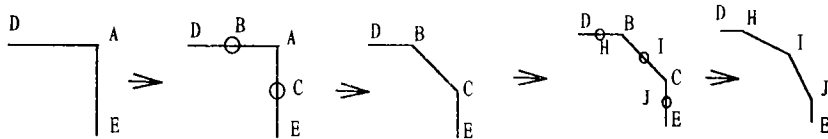


图 7 线的圆滑处理

6 拓扑关系的建立与属性值的回填

经过上述步骤, 仅仅形成了矢量的线, 而面状图的拓扑关系及其属性值并未转换, 在 WINGIS 中, 面状图的拓扑关系的建立是通过对上述的线(或进行过编辑修改的线)进行多边形生成来完成的。

6.1 多边形拓扑关系的建立及标志点的形成

即确定一多边形是由哪些线, 以怎样的位置关系来构成的。多边形生成基本原理如图 8。线 AB, BE, EF, FG, GA 构成一封闭多边形, 每条线都有多条后继线。假定从线 AB 开始生成, 线 AB 的后继线有 BC, BD, BE, 求它们和线 AB 的夹角(按顺时针方向), 夹角最小的后继线(即 BE)即为线 AB 在多边形 ABEFG 上的后继线, 然后再求线 BE 的后继线, 直到后继线和起始线重合为止, 就确定了多边形拓扑关系。特别情况的多边形如岛及花瓣(图 10)稍作相应的

处理即可,然后还应求出多边形的一个标志点(即内点)。一般算法如图9所示,设P为一多边形,maxy,miny分别为P的全部点的坐标 (x,y) 中的最大和最小的y值,作水平线 $y=\text{miny}+(\text{maxy}-\text{miny})/2$ 与P交于 (x_1,y_1) 及 (x_2,y_2) ,则点 $[(x_1+(x_2-x_1)/2),y]$ 即为多边形P的一个标志点。

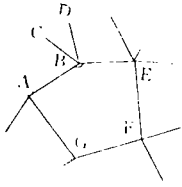


图8 多边形生成原理

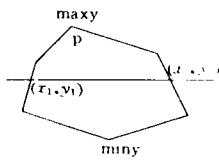


图9 多边形标志点生成

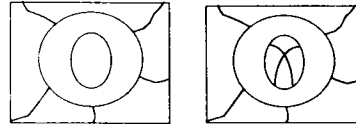


图10 特别形状的多边形岛(左)及花瓣(右)

6.2 属性回填

将多边形的标志点坐标转换成原始栅格数据的栅格点,并将该栅格点的属性值回填至矢量的多边形数据中。

通过上述方法,栅格数据已转换为矢量数据,多边形的拓扑关系已经建立,属性值已经回填,从而完成了栅格数据向矢量数据的转换。

7 结论

(1)该方法已在 WINGIS 中用 C++ 语言实现,运行可靠。通过在两个栅格点间插入边界点,并对边界点数据进行搜索成线,极大地压缩了数据。因边界点的最大方向数为 4,故每个边界点最多的搜索方向数为 4,且只要搜索点的方向数为 2,搜索就可以进行下去,由于判断条件少,从而保证了转换的快速进行。

(2)线形的处理方法简单、快速、效果理想,极大地减少了数据的冗余度。

(3)多边形拓扑关系的建立与属性值的回填是在全部线的数据形成后,采用多边形生成方法完成的。所以既保证了栅格数据向矢量数据转换的完整性,简化了转换过程,又可对线进行必要的人工编辑处理。

(4)上述方法是针对面状图的,如果稍加改动(如增加线的细化处理),也可用于线状图,如地形图的处理。

参 考 文 献

- 1 黄杏元,汤勤编著.地理信息系统概论.北京:高等教育出版社,1989.
- 2 高显连,杨国勇,孟献策,等.地理信息系统中多边形叠合算法的研究.林业科学研究,1992,5(4):442~446.

Study on the Convert of Polygon Raster Data to Vector Data in WINGIS

Li Yingguo Gao Xianlian Tian Yonglin

Abstract Technique for converting the raster data of polygon maps to vector data is discussed in this paper, based on the analysis and comparison between the raster and vector data formats. According to this technique there may be a boundary point existing between every two neighboring pixels of the raster data, that means each pixel of the raster data has at most eight boundary points with the value of 1 or 0, and the number of boundary points adjacent to another is 2 to 4. Therefore the points on a vector line and their topology can be determined because there is a number of adjacent boundary points of 3 to 4 on the nodes of a line. Through surplus point deleting, line smoothing, polygon topology generating, label point creating and feature attribute refilling, the technique can greatly decrease the data amount processed and successfully convert the raster data to vector data.

Key words geographic information system(GIS), vector data, raster data

Li Yingguo Assistant Professor, Gao Xianlian, Tian Yonglin (The Research Institute of Forest Resource Information Techniques, CAF Beijing 100091).