

空间数据的三角形二叉树结构 及其基本操作*

杨世仁 Michael F. Goodchild

(加利福尼亚大学地理系 国家地理
信息与分析中心)

Geoffreg Dutton

(Spatial Effects, 150 Irving St.
Watertown, MA 02172)

摘 要 三角形体系数据结构可建议在全球地理信息系统中使用。本文简要介绍八面体的递归细分规则及其与经、纬度之间互相换算的算法公式,地球表面上点、线和面的表示方法,以及有关三角形邻域识别、区域填充和目标膨胀的算法。

关键词 三角形体系数据结构,三角形地址,邻域识别,区域填充,目标膨胀。

1 引言

三角形体系数据结构 (THDS) 是基于一个地球状八面体的三角形二叉树分解。这种数据结构已在全球地理信息系统^[1-5]中使用。我们已建立三角形位置与笛卡尔坐标之间的转换公式、椭球地球表面上三角形面积的计算以及邻域三角形查寻的算法。文章将依次介绍用 THDS 数据结构来描述点、线和面,区域的彩色填充,椭球地球表面上区域面积的计算以及点、线、面的膨胀算法。本文介绍的算法及实验结果都将说明在 THDS 数据结构中,这些 GIS 的基本操作都可很好地实现。

2 点和线的表示

在三角形体系数据结构中,点的位置是由分解三角形的重心来标识的。三角形分解级数越高,点位置的精度越高。如在第 k 级分解中,三角形位置 A 表示为:

$$A = a_0 a_1 a_2 a_3 \cdots a_k$$

这里 a_1 至 a_k 是四进制数字, a_0 是八进制数字。如图 1 所示,我们把地球分成 8 个等边球形三角面,八进制的 a_0 值分别代表这些球形三角面的位置。

在 0 分区分中,经/纬度 (λ, φ) 与笛卡尔坐标 (x, y) 的关系如下:

$$x = \frac{2^n}{\pi} \left[\phi + 2\lambda \left(1 - \frac{2}{\pi} \phi \right) \right], \quad y = \frac{2^n \sqrt{3}}{\pi} \phi \quad (1)$$

或

* 国家地理信息与分析中心获美国国家自然科学基金,项目 SES 88-10917 支持。

本文由中国科学院遥感应用研究所图像处理室郭军和虞芳芳翻译,李小文、张保钢和朱凌校稿,最后由虞芳芳整理成论文。

收稿日期: 1995 年 1 月 24 日

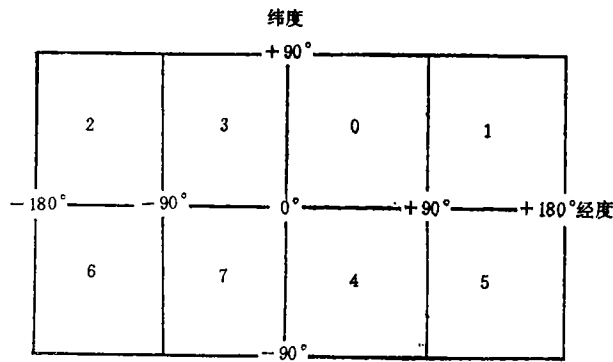


图 1 地球表面正八面体的 0 级分解

Fig. 1 Initial octahedral decomposition of the earth at level 0

$$\lambda = \frac{\pi}{2^{n+1}} \frac{\sqrt{3}x - y}{\sqrt{3} - 2^{1-n}y}, \quad \phi = \frac{\pi}{2^n \sqrt{3}} y \quad (2)$$

其中 n 是三角形分解的最大级数。

要把三角形地址转化为笛卡尔坐标，首先要把三角形地址 $a_i (i = 1, \alpha, \dots, k)$ 转化成 0 分区中的笛卡尔坐标，然后用公式(2)计算出经/纬度值，并根据 a_0 值确定分区。在 0 分区中三角形地址转换笛卡尔坐标的表达式如下：

$$x_k = \left[2^k + \sum_{i=1}^k [(-1)^{a_{i2}+1} a_{i1} 2^{k-i}] \right] 2^{n-k-1} \quad (3)$$

$$y_k = \frac{2^k + \sum_{i=1}^k [(-1)^{NZ_i+a_{i1}} [2(\bar{a}_{i1} \cap a_{i2}) + a_{i1}] 2^{k-i}]}{\sqrt{3}} 2^{n-k-1} \quad (4)$$

这里， (x_k, y_k) 是在第 k 级分解时的三角形地址的笛卡尔坐标。

其中
$$NZ_k = \sum_{i=1}^{k-1} [\bar{a}_{i1} \cap \bar{a}_{i2}]$$

是 a_1 到 a_{k-1} 中零的总数。 a_{i1} 和 a_{i2} 是表示四进制数 a_i 的两个二进制数。 \bar{a}_{i1} 和 \bar{a}_{i2} 是 a_{i1} 和 a_{i2} 的逻辑反。对于一给定经度和纬度的点，可用递归逼近处理方法^[6]来得到三角形 a_i 到 a_k 的地址值。

3 THDS 数据结构中的邻域三角形

在 THDS 数据结构中，我们把共享一条边的邻域称为直接相邻邻域，共享顶点的邻域称间接相邻邻域。现在已有在一个分区内搜寻一个三角形 3 个直接邻域的算法^[6]。对全球 GIS 的数据结构而言，需要考虑位于两个相邻分区内的相邻三角形。一个三角形根据它所在的位置，可以分成 5 类：内部三角形、边缘三角形、子边缘三角形、角三角形和子角三角形。如图 2 所示

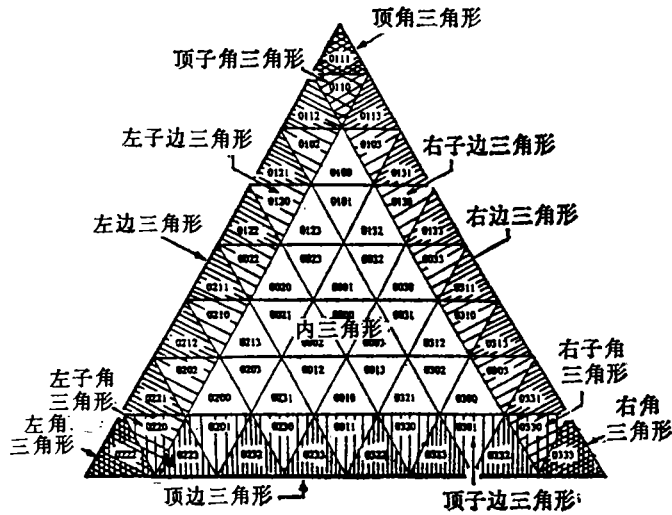


图 2 正八面体中的内、边和角三角形

Fig. 2 The inside, edge and corner triangles in an octant

三角形的邻域关系可以很容易地由它的地址编码方式确定出来(不考虑 a_0)，如图 3 所示：

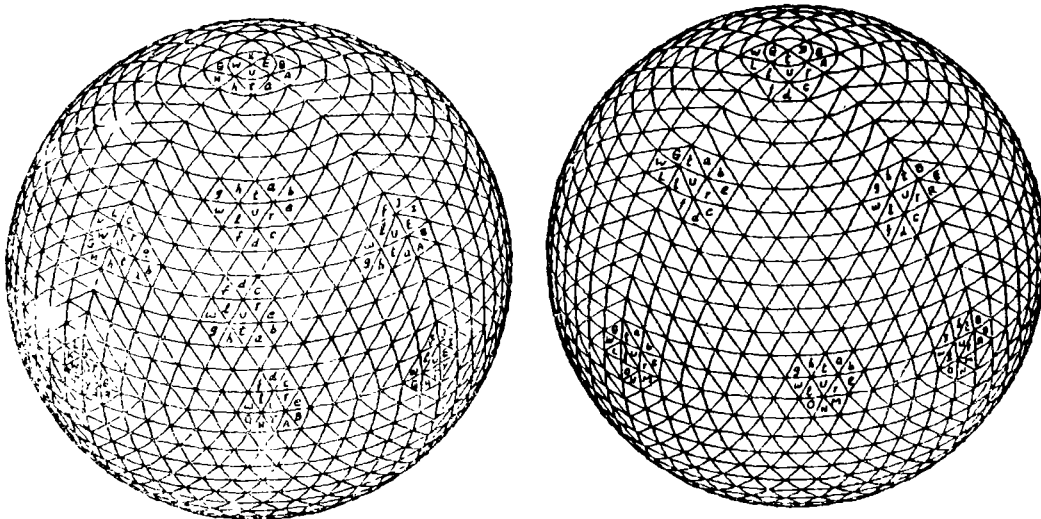


图 3 三角形位于不同位置时的邻域三角形

Fig. 3 The neighbors of a triangle when it is in different locations

内、边和角三角形的邻域

(a) The neighbors of inside, edge and corner triangles

子、边和子角三角形的邻域

(b) The neighbors of sub-edge and sub-corner triangles

我们用 t 、 l 和 r 来分别表示分区内一个三角形的顶部、左边和右边 3 个直接相邻的三角形， T 代表顶边三角形它在相邻分区的直接邻域， W 代表左边三角形在相邻分区中的

直接邻域, 代码 E 表示右边三角形在相邻分区中的直接邻域。直接邻域可以由以下算法找到:

对于给定的一个三角形 u , 把它的地址分别拷贝给它的直接邻域:

$$\begin{aligned}t &= t_1, t_2, t_3, \dots, t_k \\l &= l_1, l_2, l_3, \dots, l_k, \\r &= r_1, r_2, r_3, \dots, r_k, \\T &= T_1, T_2, T_3, \dots, T_k, \\E &= E_1, E_2, E_3, \dots, E_k, \\w &= w_1, w_2, w_3, \dots, w_k\end{aligned}$$

(1) 由于 t 和 u 在同一个分区内, t_0 值不变。当 $i = k$ 时:

如果 $t_i = 2$, 则 $t_i = 2$

如果 $t_i = 3$, 则 $t_i = 3$

并且处理 t_{i-1} 。若当 $t_i \neq 2$ 并且 $t_i \neq 3$, 那么

如果 $t_i = 0$, 则 $t_i = 1$

如果 $t_i = 1$, 则 $t_i = 0$

查找到结束。

(2) 若 l 和 u 在同一分区内, l_0 值不变。对于 $i = k$

如果 $l_i = 1$, 则 $l_i = 3$

如果 $l_i = 2$, 则 $l_i = 1$

并且处理 l_{i-1} 。若当 $l_i \neq 1$ 并且 $l_i \neq 2$, 那么

如果 $l_i = 0$, 则 $l_i = 2$

如果 $l_i = 3$, 则 $l_i = 0$

查找到结束。

(3) 对于同一分区内的 r 和 u , r_0 值不变。对于 $i = k$

如果 $r_i = 1$, 则 $r_i = 2$

如果 $r_i = 3$, 则 $r_i = 1$

并且处理 r_{i-1} 。若 $r_i \neq 1$ 并且 $r_i \neq 3$

如果 $r_i = 0$, 则 $r_i = 3$

如果 $r_i = 2$, 则 $r_i = 0$

查找到结束。

(4) 对于位于顶部相邻分区中的直接邻域 T ,

如果 $T_0 \leq 3$, 那么 $T_0 = T_0 + 4$

如果 $T_0 \geq 4$, 那么 $T_0 = T_0 - 4$

T_i (i 从 1 到 k) 值不变。

(5) 左边分区中的直接邻域有:

如果 $w_0 \neq 0$ 或 $w_0 \neq 4$, 那么 $w_0 = w_0 - 1$

如果 $w_0 = 0$ 或 $w_0 = 4$, 那么 $w_0 = w_0 + 3$

如果 $w_i = 2$ ($i = 1, \dots, k$), 那么 $w_i = 3$

(6) 位于右边分区中的直接邻域有:

如果 $E_0 \neq 3$ 或 $E_0 \neq 7$, 那么 $E_0 = E_0 + 1$

如果 $E_0 = 3$ 或 $E_0 = 7$, 那么 $E_0 = E_0 - 3$

如果 $E_i = 3(i = 1, \dots, k)$, 那么 $E_i = 2$

如图 3 所示, 在分区内三角形在不同的位置上, 它的间接邻域的特点也不同, 它们可由三角形的直接邻域地址值来确定。

4 THDS 结构中线的表示

由于三角形地址与经/纬度(或笛卡尔坐标)可以相互转换, 用 THDS 结构查找一个三角形的邻域非常有效。可按下列步骤进行线段插值:

(1) 把三角形地址转换到经/纬度, 用线性或 B 样条插值函数进行插值。

(2) 把经过插值的经/纬度值转换回三角形地址。

(3) 当依次的两个三角形不相邻时, 不断对它们的经/纬度进行插值, 直到转换回的两个连续三角形地址相邻或相同。

(4) 如果相邻三角形具有相同的地址, 则去掉重复的地址。

在计算机制图和地理信息系统中的数据压缩技术中已广泛使用链码来表示线段^[7]。

在 THDS 中最简单的链码是 3-code 链码, 它连接相邻两个直接邻域三角形的重心。但是 3-code 太不规则而且它不能连接位于不同分区中的三角形。因此我们建议在 3-code 链码上再加以三角形东西两个间接邻域三角形 e 和 w 以及位于相邻分区的 T 、 E 和 W 三角形代码, 这样形成 8-code (i, l, r, e, w, T, E, W) 来表示区域的边界线。

当数据精度降低时, 线段会显示出不必要的“锯齿”甚至尖峰。尖峰需要被去除, 锯齿需要平滑掉。“锯齿”状线段的链码表示为 $lll \cdots l$ 或 $rrr \cdots r$, 在相应的位置上替换成 $ww \cdots w$ 或 $ee \cdots e$ 就可平滑“锯齿”线段。尖峰的链码为 tt 或一系列连续 $\cdots lrlrlr \cdots$, 只要把这些代码去掉就可消除尖峰的影响。

5 THDS 结构中的区域填充

我们用均等检验算法来确定给定边界线的区域。区域填充是从低纬向高纬一行一行进行, 每一行中, 由西向东填充。算法包括以下几个步骤:

(1) 给每个边界三角形赋一个标志。这个标志由点与其前趋和后继的关系而定。共有 5 种不同的标志。

标志 a 赋给没有东或西邻域三角形的点。它的前趋三角形是顶邻域三角形并且后继三角形是左或右邻域三角形, 或者它的前趋三角形是一个左或右邻域三角形并且它的后继是一个顶邻域三角形。我们称这些点为单交叉点。

标志 b 赋给没有西邻域三角形的点。如果该三角形是朝向上的, 那么它的前趋三角形是一个顶邻域三角形并且它的后继三角形是一个东邻域三角形。如果该三角形方向向下, 那么它的前趋三角形为左或右邻域三角形并且它的后继三角形是一东邻域。我们称

这些点为向上切线入口点。

标志 *c* 赋予无西邻域三角形的点。如果该三角形向上，它的前趋三角形是在左或右邻域。它的后继三角形在东邻域。如果该三角形向下，那么它的前趋三角形是一顶邻域三角形并且它的后继三角形是一东邻域。我们称这些为向下切线入口点。

标志 *d* 赋予一个无东邻域的点，如果给定三角是向上，它的前趋三角形在西邻域且它的后继三角形是顶邻域。如果该三角形向下，那么它的前趋三角形是一西邻域且它的后继三角形是一左或右邻域。我们称这些为向下切线退出点。

标志 *e* 赋予无东邻域的点，如果给定三角形向上，那么它的前趋三角形是西邻域并且它的后继三角形是左或右邻域。如该三角形向下，它的前趋三角形是西邻域并且它的后继三角形是顶点邻域。我们称之为向上切线退出点。在图 4 和图 5 中列出这些点及其表现形式。

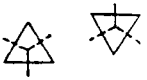
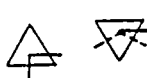
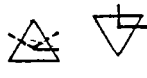
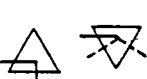
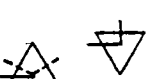
模式	点的类型	判别式	
		向上三角形	向下三角形
a 	单交叉点	$i_{i-1} \cap (i_{i+1} \cup \bar{i}_{i+1}) \cap \bar{e}_{i+1} \cap \bar{w}_{i+1}$ or $i_{i+1} \cap (i_{i-1} \cup \bar{i}_{i-1}) \cap \bar{e}_{i+1} \cap \bar{w}_{i+1}$	
b 	向上正切入口点	$i_{i-1} \cap \bar{i}_{i+1} \cap \bar{e}_{i+1} \cap \bar{w}_{i+1}$	$\bar{i}_{i-1} \cap (i_{i-1} \cup \bar{i}_{i-1}) \cap \bar{e}_{i+1} \cap \bar{w}_{i+1}$
c 	向下正切退出点	$i_{i-1} \cap (i_{i+1} \cup \bar{i}_{i+1}) \cap \bar{e}_{i+1} \cap \bar{w}_{i+1}$	$\bar{i}_{i-1} \cap i_{i-1} \cap \bar{e}_{i+1} \cap \bar{w}_{i+1}$
d 	向下正切退出点	$i_{i+1} \cap \bar{i}_{i+1} \cap \bar{e}_{i-1} \cap \bar{w}_{i-1}$	$\bar{i}_{i-1} \cap (i_{i+1} \cup \bar{i}_{i+1}) \cap \bar{e}_{i-1} \cap \bar{w}_{i-1}$
e 	向上正切退出点	$\bar{i}_{i-1} \cap (i_{i+1} \cup \bar{i}_{i+1}) \cap \bar{e}_{i-1} \cap \bar{w}_{i-1}$	$i_{i+1} \cap \bar{i}_{i+1} \cap \bar{e}_{i-1} \cap \bar{w}_{i-1}$

图 4 区域边界的三角形的标志设置

Fig. 4 Flags assigned to boundary triangles in the region filling algorithm. In the notation used to represent the region boundary using 8-code, it is true if the *i*-th entry in the chain is move to the top neighbor, etc.

(2) 将三角形地址转换成经/纬度 $\phi[i]$ 和 $\lambda[i](i = 1, 2, \dots, n)$, 并且按 ϕ 的升序排列。这里 n 是由 8-code 链码表示的边界点的数目。

点的类型	区域填充时边界线穿过三角形中心点的类型
简单点 (非交叉点)	
向上正切入口 向下正切出口(非交叉点)	
向下正切入口 向上正切出口(非交叉点)	
单点交叉入口或出口 (可能是交叉点)	
向上正切入口 向上正切出口 (可能是交叉点)	
向下正切入口 向下正切出口 (可能是交叉点)	

图 5 边界三角形类型

Fig. 5 Types of boundary triangles

(3) 对给定的 ϕ 按 λ 的升序排列该数据矩阵。我们将使用一对二维矩阵 $A[i][j]$ 和 $flag[i][j]$ 。下标 i 表示由区域最底行起计算的行数(或扫描行数); 下标 j 表示第 i 行点的数目。

(4) 为使具有复杂拓扑结构的区域能被正确填充, 我们用两个计数器 n_1 和 n_2 。它们在初始行被赋零, 然后,

- 对每一个向上的切线入口点, n_1 增加 1。
- 对每一个向下的切线入口点, n_2 增加 1。
- 对每一个向下的切线退出点, n_1 减少 1。
- 对每一个向上的切线退出点, n_2 减少 1。

(5) 用一个奇/偶标志 eo 表示在一条扫描线上两点之间的线段是否落在区域之内。在初始行, $eo = 0$, 在遇到下列情况时, 我们将 eo 增加 1:

- 一个单交叉点。
- 一个向下的切线退出点, 它的 $n_1 \geq 1$ 。
- 一个向上切线退出点, 它的 $n_2 \geq 1$

(6) 把区域或孔洞的边界点复印到对应行, 使它们包含在这个区域之内。

(7) 如果 e_0 标志为偶数, 并且该点所在三角形地址不等于它的后继三角形地址, 那么用西邻域来填充扫描行。

6 球形或椭球形地球表面上的区域面积

对于球形地球, 当分解级较高时, 球面上三角形面积可用下式来计算^[6]

$$\Delta A_\phi = k \frac{\cos \phi}{\frac{\pi}{2} - \phi} = k \frac{\sin \chi}{\chi} = k \operatorname{sinc} \chi$$

这里

$$\operatorname{sinc} \chi = \frac{\sin \chi}{\chi}, \quad k = \frac{\pi^3 R^2}{2^{2n+4}}, \quad \chi = \frac{\pi}{2} - \phi. \quad (6)$$

R 是地球平均半径, n 为分解级。

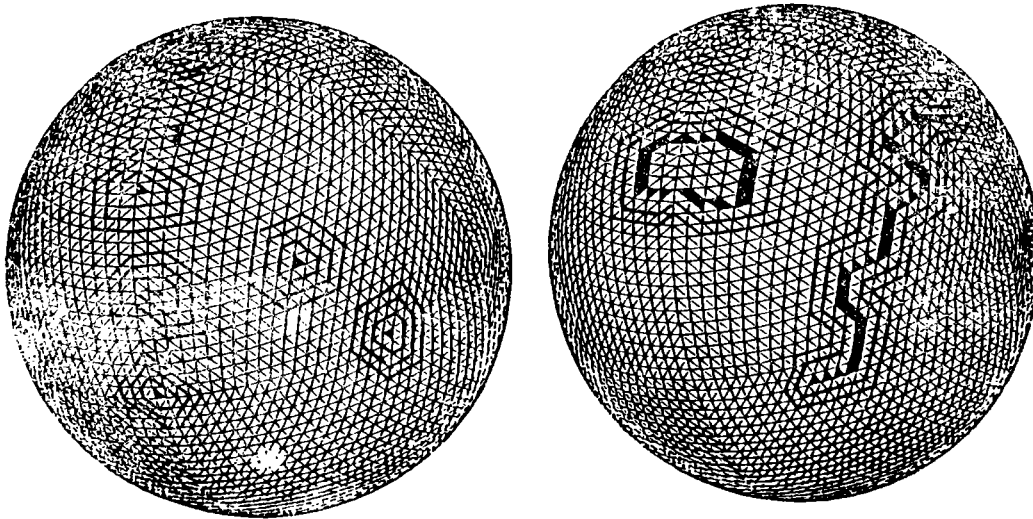
对于椭球形地球, 对于分解级为 k 的三角形, 面积计算表达式为:

$$\Delta A_\phi = k \left[\operatorname{sinc} \chi + \frac{r^2}{8R^2} \operatorname{sinc} \chi + \frac{r^2}{8R^2} \frac{\sin 3\chi}{\chi} \right]. \quad (7)$$

R 是椭球长轴, r 是椭球长短轴之比。

7 THDS 结构中的膨胀算法

在 GIS 中, 膨胀也叫“缓冲 (buffering)”。第 3 节中描述的邻域三角形搜寻算法可直接用于膨胀运算, 因为膨胀也需要寻找点、线或面的所有相邻要素。



(a) 点的膨胀 (b) 线和区域的膨胀

图 6 点、线和面的膨胀

Fig. 6 Dilation of point, line and region

用邻域搜寻算法识别出所有与目标点(线、面)相连的直接或间接三角形,去掉重复的三角形,扩大该区域,直到膨胀到要求的范围。对于内三角形,膨胀区域如果没有和另一分区相交,那么它是一个六边形,六边形其中 3 条边长是 $(m-1) \times l$,另外 3 条边长是 $m \times l$,这里 m 是重复扩张过程的次数, l 是指定分解级中三角形的边长。如果区域跨过分区边界,区域将会变形,如图 6 所示。

8 高层定位

前面我们讨论的都是二维的三角形,本节中我们将检查八面体分区的三角形顶点及其分解三角形顶点的位置属性以及它们对 THDS 空间数据操作的潜在影响。

当一个三角形分解时,形成 3 个“外部”瓦片和一个“内部”瓦片。每个外部瓦片都与父三角形的一个顶点相联系。反过来,对于每个父顶点,它有 6 个子瓦片相连(被包围)(除八面体的 6 个原始顶点外,它们每个被 4 个瓦片包围)。当分解级较高时。我们可以用这个顶点来表示所有在它周围 6 个瓦片上的所有点,就象用三角形地址来表征所有位于三角形内的点一样。Dutton^[4]称这种顶点为“吸子”。在描述空间邻域的关系方面,吸子要比用三角形地址方法方便得多。

吸子地址的计算算法包括瓦片地址与 Putton (或 Fekete^[8]) 地址之间转换算法以及第 3 节中描述的邻域搜寻算法。比较而言,THDS 结构的地址编码模型在数据编码以及瓦片操作方面计算起来更有效。

9 结论

THDS 数据结构模型是由三角形组成的区域四叉树,这些三角形聚集形成六边形。它用线性的地址来表示空间球形地理数据,并且提供在任一比例尺体系的球体上任一位置的连续地学参数。前面已描述了 THDS 的各个特征:瓦片编号,相邻邻域的认识,链码编号,区域填充,面积计算,膨胀过程以及“吸子”的高层结构。“吸子”结构有助于处理区域四叉树中相邻子树的关系,可避免在 THDS 领域中进行叠合地图特征操作时形成多边形“薄条”。

参 考 文 献

- [1] G. Dutton, Geodesic modeling of planetary reliefs, *Cartography*. 1984, 21: 188—207.
- [2] G. Dutton, Planetary Modeling via hierarchical tessellation. *Proceedings. AutoCarto ASPRS/ACSM Fall Church VA 1989*, 462—471.
- [3] G. Dutton, Modelling locational uncertainty via hierarchical tessellation, S. Gopal, eds., *Accuracy of Spatial Database* London, Taylor and Francis. 125—140, 1989.
- [4] G. Dutton, Locational properties of Quaternary Triangular Meshes, *Proceedings, Fourth International Symposium on Spatial Data Handling* University of Zurich, 1990.
- [5] G. Dutton, Polyhedral hierarchical tessellations: the shape of GIS to come. *Geo Info Systems* 49—55, February, 1991.
- [6] M. F. Goodchild and Yang Shiren. A hierarchical spatial data structure for a global geographic information system. To appear in *Computer Vision, Graphics and Image Processing*.

- [7] H. Freeman. On the encoding of arbitrary geometric configurations. *Transactions on Electronic Computers*. 1961, EC10: 260—268.
- [8] G. Fekete. Rendering and managing spherical data with sphere quadtrees. *Proceedings of Visualization 90 Association for Computing Machinery*. New York, 1990.

Spatial Data Representation and Basic Operations for a Triangular Hierarchical Data Structure

Yang Shiren Michael F. Goodchild

(*University of California at Santa Barbara Na-
tional Center for Geographic Information and
Analysis and Department of Geography*)

Geoffrey Dutton

(*Spatial Effects, 150 Irving St, Wa-
tertown, MA 02172*)

Abstract A triangular hierarchical data structure has been proposed as the basis for a global geographical information system. In this paper we briefly review one such scheme based on recursive subdivision of an octahedron, and conversion algorithms to and from latitude/longitude. Schemes for representing point, line and area objects on the earth's surface are described. We present algorithms for identifying triangle neighbors, region filling and object dilation.

Key words Triangular hierarchical data structure, Triangular address, Neighbor identification, Chain encoding, Region filling, Dilation