

基于四叉树结构的数字地表模型快速生成算法设计

谢传节¹⁾ 万洪涛²⁾

¹⁾(中科院地理科学与资源研究所资源与环境信息系统国家重点实验室, 北京 100101)

²⁾(中国科学院遥感应用研究所, 北京 100101)

摘要 为了研究数字地表模型的快速生成方法,在总结传统 Delaunay 三角化算法的基础上,给出了一个基于四叉树结构的数字地表模型快速生成算法的详细设计.该算法的基本思想是首先利用四叉树结构来对离散点进行分割,然后对四叉树叶节点进行 Delaunay 三角化,再两两合并四叉树节点三角网的凸壳,以快速生成地表格网模型.该算法是以四叉树为基本单位来实现限定边和限定多边形的快速嵌入;最后给出了算法在不同情况下的测试结果,并对测试结果进行了具体分析,给出了算法的时间效率分析和空间复杂性分析.实测数据结果表明,该算法有着较好的性能,而且也非常稳定.通过实测结果分析和算法的时间效率分析,可以得到算法的时间效率近似为 $O(n\log(n))$;通过算法的空间复杂性分析可以看出,算法可以自动适应不同的点空间分布情况,而且采用四叉树结构也非常有利于限定边和限定多边形的嵌入.

关键词 地理信息系统 数字地表模型 限定 Delaunay 三角化

中图法分类号: P208 文献标识码: A 文章编号: 1006-8961(2002)04-0394-06

Algorithm for Rapidly Generating Digital Terrain Model Based on Quad-tree

XIE Chuan-jie¹⁾, WAN Hong-tao²⁾

¹⁾(State Key Lab. of Resources and Environment Information Systems,
The Institute of Geography Science and Resources, CAS, Beijing 100101)

²⁾(Institute of Remote Sensing Applications, Chinese Academy of Sciences, Beijing 100101)

Abstract In this paper, an algorithm to generate digital terrain model rapidly is presented. The algorithm is based on the constrained delaunay triangulation(CDT) algorithm. The algorithm achieves a high performance by efficient managing data for the digital terrain model by quad-tree, and by reducing the calculation of the algorithm by quad-tree. At first, the discrete points of the digital terrain model are distributed to different leaf nodes of the quad-tree; then, the points in the leaf node of quad-tree are triangulated by delaunay triangulation algorithm; at Last, meshes in the nodes which are neighborhood in space are coalesced together to generate a new smooth mesh. The algorithm is high performance because in the coalition which only need to deal with the points in the hull of the mesh in each node of quad-tree. Moreover, the constrained edges and constrained polygons can be integrated to the mesh quickly by using quad-tree structure. In the paper, the test results of the algorithm and the analysis to the test results are also presented. Moreover, a test picture of the algorithm is illustrated. In the last part of the paper, the time cost analysis of the algorithm and the spatial characteristics analysis of the algorithm are presented. The test results show that the algorithm not only has excellent performance but also is robust. The time cost analysis of the algorithm shows that the expected time of the algorithm is $O(n\log(n))$, and the analysis of the spatial characteristics of the algorithm shows that algorithm can be adaptive to different spatial distributed situation of the points set to be triangulated. Meanwhile, the constrained edges and the constrained polygons can be integrated into the triangulated mesh efficiently with the help of quad-tree structure of the algorithm.

Keywords Geographical information system, Digital terrain model, Constrained Delaunay triangulation

0 引言

数字地表模型 DTM 是构成地形表面形态多种信息的一个数字表示。一般数字地表指的是以构成地表的特征点为基础,由描述地表的特征点、特征线和特征面共同构成的数字地形面,数字地表模型生成算法则用于将构成数字地形面的特征点、特征线和特征面结合到一起来生成数字地表模型(DTM),而数字地表模型生成的关键是三角化算法。国内外对三角化生成算法的研究很多,已提出了多种不同的三角化方法^[1-6],其中, Delaunay 三角化算法已经得到普遍认可,当前对数字地表模型生成算法研究,更多的是研究如何利用不同的方法和不同的数据结构来提高 Delaunay 算法的效率,如今已形成了不同类型的 Delaunay 算法。本文设计的数字地表模型快速生成算法是以限定 Delaunay 三角化(Constrained Delaunay Triangulation)算法为基础,通过采用四叉树结构来管理数据,并以四叉树结构为基础来设计算法,以便获得较高的算法执行效率。

1 Delaunay 三角化算法研究现状

Delaunay 三角化算法起源于计算几何学中 Voronoi 图方面的研究,是当前主要的三角网生成算法,经过 20 世纪 70 年代以来的发展,已形成了多种 Delaunay 三角化算法,但普遍采用的主要有分割-归并算法(Divide-and-Conquer)^[7,8],逐点插入法^[9,10]和三角网生长法^[11,12],现简介如下:

(1) 分割-归并算法的基本步骤是首先对点集进行分割,即把点集划分到足够小,以利于对每个分割出来的子集进行 Delaunay 三角化,然后再合并相邻子集的凸壳,进而生成整个点集的三角网模型。另外,分割-归并算法按分割方法的不同,可以分为条带分割方法,网格分割方法和四叉树分割方法。

(2) 逐点插入法是 Delaunay 三角化算法中步骤最为简单的一种,其基本方法是首先定义一个包含了点集所有数据点的初始多边形,一般为点集的凸壳;然后对初始多边形进行 Delaunay 三角化,形成初始三角网;最后向已建立的三角网中,逐点插入点集中的那些不包含在已经建立的三角网中的点,直至形成最终包含点集中所有点的三角网。逐点插入法的关键是快速确定插入点在已建立的三角网中

的位置。

(3) 三角网生长法的基本步骤是首先选定点集中一点为起始点,再找出与起始点最近的一点,并相互连接以形成基线边;然后按照 Delaunay 法则,从点集中找出第 3 点,并与基线边一起来形成三角形,同时生成新的基线边;最后以新生成的基线边为基础,按照上述方法向三角网中添加三角形,直至所有的基线边都被处理为止。三角网生长法的关键是对于一条基线边,如何快速确定第 3 点。

本文在设计算法时,因为不但要考虑三角网本身生成的效率,同时也要考虑数字地表模型中构成数字地形面的特征线和特征面插入的效率,所以采用了基于四叉树分割的分割-归并算法的基本思想来设计数字地表模型快速生成算法。

2 算法的设计

算法的执行过程可以分为数据的预处理、地表三角网生成和特征线和特征面的嵌入 3 个步骤。

2.1 数据的预处理

数据预处理的主要任务是要执行必要的检查,以保证算法的稳定性,同时将原始特征点、特征线和特征面,按四叉树结构组织起来。数据预处理过程如下:

(1) 首先对特征线和特征面进行必要的检查,合并特征线或特征面上一些边长小于设定最小边长值的超短边;

(2) 将特征点和处理后的组成特征线、特征面的点组合到一起,删除坐标值出现重复的点和点间距离超过点间距离最小值的非关键点;

(3) 在处理后的点的集合上,生成点的四叉树结构,并设定每个四叉树结构叶节点中可容纳的最多点数。四叉树叶节点中点数(四叉树结构叶节点中最多可容纳的点数)虽可以采用常用的分割方法分割到不超过 3 个,但这样大量节点合并,需要占用大量的时间。考虑到内存空间效率和后面特征线和特征面插入的需要,四叉树结构叶节点中最大点数的建议值为 16 个点;

(4) 将特征线以边结构表示,并将特征面转化为边的拓扑结构,再以边为基本单位,将特征线和特征面信息表达达到步骤(3)中点的四叉树结构中,以形成地表数据的四叉树结构表达。

2.2 地表三角网生成

所谓地表三角网生成就是将地表数据二叉树结构中的点,快速连接为地表三角网.地表三角网生成过程可以分为二叉树结构叶节点三角网的生成和二叉树节点间三角网的合并两大步骤.其中,二叉树结构叶节点三角网的生成采用的是基本的 Delaunay 三角化方法,即首先为地表数据的二叉树结构叶节点中的点生成凸壳(shell);然后将凸壳按 Delaunay 法则连接成三角网;再将叶节点中的非凸壳点插入到以凸壳为边界的三角网中,以完成二叉树叶节点的三角化.由于一般的 Delaunay 三角化方法已在很多文献中讲述,所以这里不再介绍.

由于二叉树结构叶节点中的点可能少于3个或位于同一条线上,不能构成三角网,所以节点间的凸壳合并存在点和凸壳的合并和凸壳的两两合并两种情况.

2.2.1 点和凸壳的合并

点和凸壳的合并是将点加入到凸壳之中,同时将凸壳上部分边转化为非凸壳边,但这种点和凸壳的合并与点和凸壳的位置有关.下面以点在凸壳的左边为例来介绍点和凸壳的合并过程,其他情况类似.

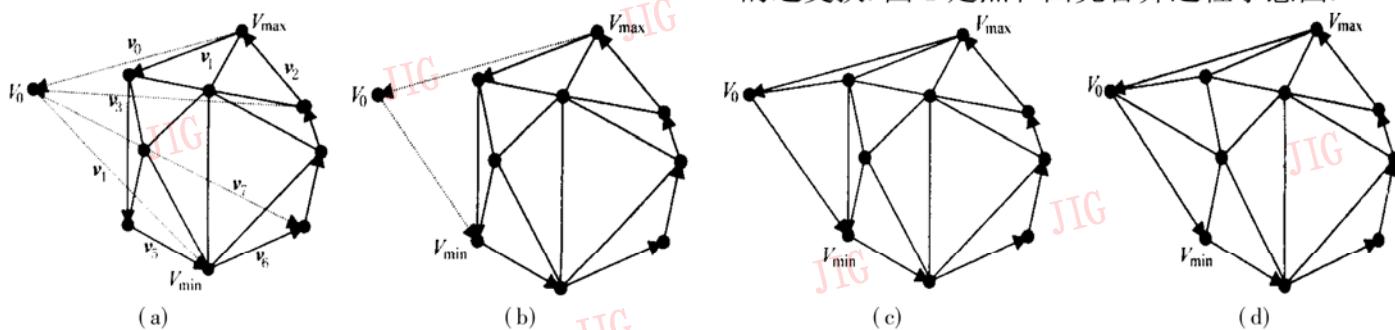


图1 点和凸壳合并过程示意图

2.2.2 凸壳的两两合并

凸壳的两两合并分为凸壳的左右合并和凸壳的上下合并两种情况.下面以凸壳的左右合并为例,来简述凸壳的两两合并的过程,而凸壳的上下合并与左右合并相类似.

(1) 首先求取左、右凸壳在 y 方向上的最大值点 $V_{\max 1}$ 、 $V_{\max 2}$, 和最小值点 $V_{\min 1}$ 、 $V_{\min 2}$, 如图 2(a) 所示.

(2) 按照点和凸壳合并类似的方法移动点 $V_{\max 1}$ 、 $V_{\max 2}$ 、 $V_{\min 1}$ 、 $V_{\min 2}$, 直到这些点不在需要移动为止;

(3) 合并左、右两凸壳,使 $V_{\min 1}$ 到 $V_{\max 1}$ 之间的

(1) 寻找凸壳上 y 方向坐标值最大的点 V_{\max} 和 y 方向坐标值最小的点 V_{\min} ;

(2) 如图 1(a) 所示,定义 V_{\max} 到 V_0 方向矢量为 v_0 ,以 V_{\max} 为起点的方向矢量为 v_1 ,以 V_{\max} 为终点的方向矢量为 v_2 ,由 v_2 起点到 V_0 的方向矢量为 v_3 .合并时,如 v_0 在 v_1 的左侧(矢量方向的左侧),则 V_{\max} 点要前移到 v_1 的终点;如 v_0 在 v_1 的右侧,则比较 v_2 与 v_3 之间的关系,如 v_3 在 v_2 的右侧,则 V_{\max} 点要后移到 v_2 的起点;

(3) 重复步骤(2)直到 V_{\max} 不再需要移动为止;

(4) 如图 1(a) 所示,定义 V_0 到 V_{\min} 的方向矢量为 v_4 ,以 V_{\min} 为终点的方向矢量为 v_5 ,以 V_{\min} 为起点的方向矢量为 v_6 , V_0 到 v_6 终点的方向矢量为 v_7 .如 v_4 在 v_5 的左侧,则 V_{\min} 点要后移到 v_5 的起点;如 v_4 在 v_5 的右侧,则比较 v_6 与 v_7 之间的关系,如 v_7 在 v_6 的右侧,则 V_{\min} 点要前移到 v_6 的终点;

(5) 重复步骤(4),直到 V_{\min} 不再需要移动为止;

(6) 合并 V_{\max} 到 V_{\min} 之间的凸壳边,构成新的三角形,并以 V_{\max} 到 V_0 之间的方向矢量和 V_0 到 V_{\min} 之间的方向矢量为新的凸壳边.在构成新的三角形时,一定要执行 Delaunay 准则检查,进行必要的边交换.图 1 是点和凸壳合并过程示意图.

方向矢量和 $V_{\max 2}$ 到 $V_{\min 2}$ 之间的方向矢量成为新凸壳内部一些新的三角形的边,同时要以 Delaunay 准则检查新生成的三角形,然后以 $V_{\max 2}$ 到 $V_{\max 1}$ 之间的方向矢量和 $V_{\min 1}$ 到 $V_{\min 2}$ 之间的方向矢量为新的凸壳边.

图 2 是凸壳的两两合并过程简单示意图.

2.3 特征线和特征面的嵌入

在完成地表三角网生成后,就要把地表特征线(如地形线、断裂线等)和地表特征面(如外边界多边形,内部剪切多边形等)嵌入到地表模型中.由于特征面也被转化为边的拓扑结构,所以,特征面首先也是作为特征线的一种,插入到地表三角网中,然后再

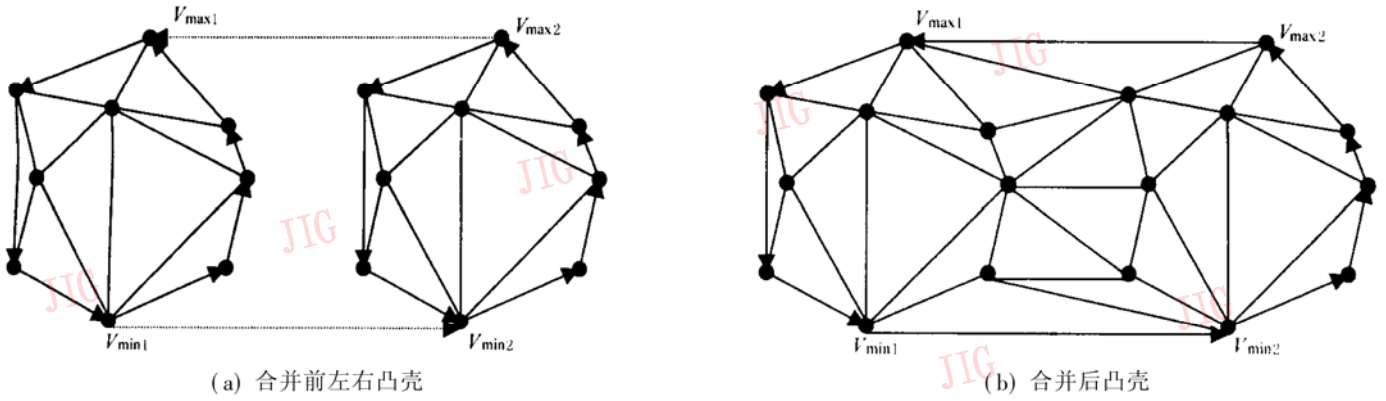


图 2 凸壳合并过程示意图

作专门的特征多边形处理. 在特征线和特征面的嵌入过程中, 应充分利用四叉树结构和特征线拓扑结构, 以获取较高的特征线和特征面嵌入效率.

2.3.1 特征线的插入

由于特征线上的点已经作为三角网的基本点存在于三角网中, 因此特征线的插入就是通过寻找与特征线相交的三角形, 并计算特征线与这些三角形的交点, 然后把交点插入到特征线中, 同时将这些相交点插入到三角网中, 以便将特征线插入到三角网中. 这种特征线的插入是以构成特征线的线段为基本插入单位, 当所有构成特征线的线段都被插入到三角网中时, 也就完成了整个特征线的插入. 线段插入过程可以分为以下几个步骤:

- (1) 以线段的一点作为出发点, 利用四叉树结构, 快速寻找以该点为顶点, 并与构成特征线的线段相交的起始三角形;
- (2) 求出线段与相交三角形的交点, 再用交点分割线段来生成新的线段, 包括新线段的拓扑关系, 从而把该点插入到线段中;
- (3) 把交点插入到地表三角网中, 以生成新的三角形, 并按 Delaunay 法则, 检查新生成的三角形;
- (4) 取出线段的下一点, 如该点不是线段的终点, 则重复步骤 1~ 步骤 4, 否则线段插入结束.

图 3 是特征线中一个线段插入示意图.

2.3.2 特征多边形的处理

经过特征线的插入, 特征多边形边界已经存在于三角网中, 而特征多边形的处理就是根据特征面的功能来修改三角网中的三角形. 特征多边形功能可以有下面的 3 种类型:

- ① 外部边界多边形, 位于外部边界多边形以外的三角形都被裁剪掉;
- ② 内部剪切多边形, 位于内部剪切多边形内的三角形都被裁剪掉;

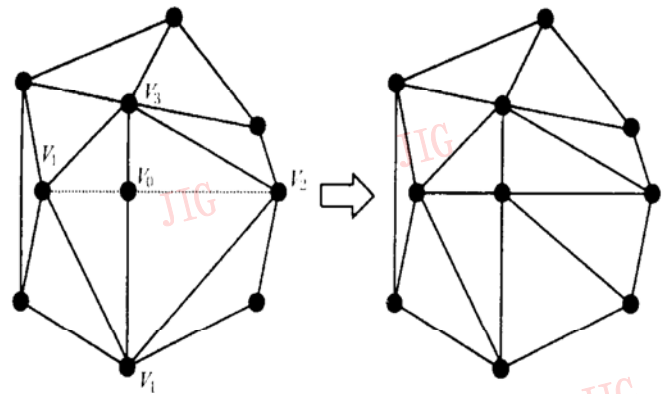


图 3 特征线插入示意图

(图中 V_1V_2 为构成特征线的线段, V_0 为交点)

③ 平面多边形(即高程值均相同的多边形), 位于平面多边形内所有点的高程值都被修改为同一个值.

这几种特征多边形的功能虽然各不相同, 但处理方法却很相似, 特征多边形的一般处理过程为:

- (1) 初始化一个边的链表, 将特征多边形各边添加到该链表中;
- (2) 从边的链表中取出一个边, 按特征多边形的功能来处理该边两侧的三角形, 删除处理完的边, 如果必要, 向边链表中添加需要处理的非边界边;
- (3) 重复步骤 2 直到边链表为空.

3 算法测试结果分析

在进行算法的测试分析时, 对比了如下 3 种基本情况: 一是不采用空间分割方法, 即使四叉树节点的最大点数超过点集中的总点数; 二是把点集中的点数设为不超过 3 个点, 这样就不存在四叉树叶节点三角网生成时间; 三是把点集中的点数设为不超过 16 个点. 实际测试所用时间如表 1 所示(所用时间仅为三角网生成时间, 不包括与三角化无关的初始化时间和特征线和特征面嵌入时间, 机器主频

为: 350MHz, 内存 128M)。

表 1 数据测试算法运行时间

点数	单位: s		
	不采用空间分割	四叉树叶节点数 不超过 3 个	四叉树叶节点数 不超过 16 个
2 500	2.04	0.51	0.44
10 000	43.31	2.02	2.03
200 000	> 1200	51.91	52.7

从表 1 的 (1)、(2)、(3) 3 种情况对比可见, 不采用空间分割和采用空间分割方法的结果差别悬殊, 特别是在点数为 200 000 时, 第 (1) 种情况在 20min 内还没有运行结束。而从第 (2) 种和第 (3) 种情况对比可见, 尽管第 (2) 种情况在理论上不存在四叉树叶节点三角网的生成时间, 但是会大大增加节点间凸壳合并的时间, 由实测结果可见, 第 (2) 种和第 (3) 种情况在时间效率上是相当的, 而第 (2) 种情况还会大大增加四叉树结构占用的内存, 同时也增加了后来特征线和特征面嵌入时, 在四叉树结构上的搜索时间。

图 4 是运用本文算法进行数据测试生成的结果示例图。

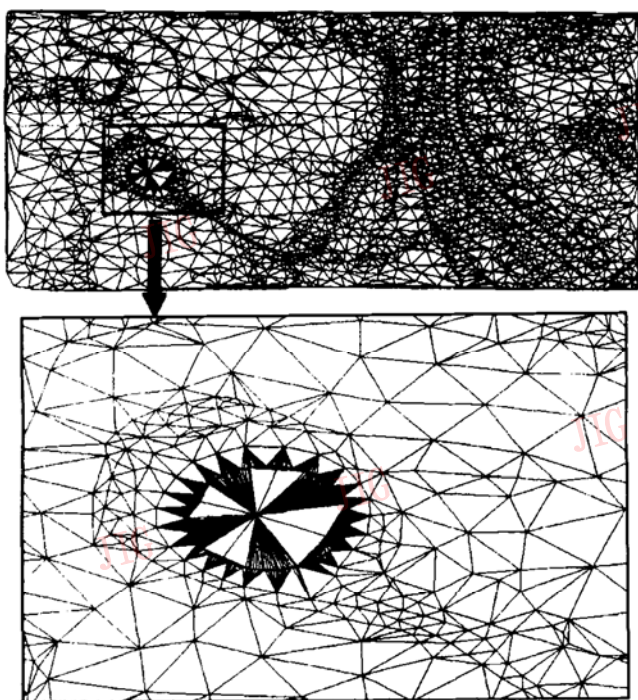


图 4 算法运行结果示例图

4 算法复杂性分析

4.1 算法时间复杂性分析

本文算法中的关键部分, 即地表三角网生成的时间复杂性分析如下: 三角网生成时间包括四叉树生成和四叉树叶节点三角化及合并时间, 假设点集

中, 离散点的总数为 n , 四叉树叶节点中最大允许点数为 k , 那么算法复杂性分析如下:

由于初始化过程有很多是附加的, 它们是用来保证数据的完整性的, 所以仅需要考虑四叉树的生成时间, 且实际运算中, 四叉树的生成过程非常简单, 其时间复杂度大致为 $O(\sqrt{n})$ 。

(1) 四叉树叶节点的三角化时间复杂性分析

假设四叉树叶节点的总数为 m , 一般 $m < 2n/k$ 。四叉树叶节点三角化的处理时间包括叶节点凸壳生成时间和其他点的插入时间。对于一个四叉树叶节点 I , 假设它的点数为 l_i , 则其凸壳生成时间复杂度为 $O(l_i)$, 其他点的插入时间复杂度总和约为 $O(l_i \times l_i)$, 由此可见, 四叉树叶节点的三角化时间复杂度总和为 $O(m \times (O(l_i) + O(l_i \times l_i)))$, 由于 l_i 不超过 k , 而 m 一般不超过 $2n/k$, 所以最坏情况下, 四叉树叶节点的三角化时间复杂度为 $O(n \times (2 + k))$ 。

(2) 四叉树叶节点合并时间复杂性分析

由于合并是两两进行, 而且只同节点凸壳上的点有关。因此最坏情况下, 四叉树叶节点合并时间复杂度为 $O(m/2 \times \sqrt{m/2})$, 若将 m 的最大值 $2n/k$ 代入, 则最坏情况下, 其时间复杂度为 $O((n/k) \times \sqrt{(n/k)})$ 。

由此可见, 最坏情况下, 整个算法时间复杂度的总和为

$$O(2n + n \times k + \sqrt{n} + (n/k) \times \sqrt{(n/k)})$$

其中, n 为总点数, k 为四叉树结构叶节点中最大允许点数。

由此可见, 算法的时间复杂度与 k 有着紧密联系, 但当 k 稳定为某一值时 (一般为 16), 算法有着相当的稳定性。实际测试所需时间如表 1 所示。

由以上分析可见, 算法实际的时间效率 (复杂度) 近似为 $O(n \log(n))$ 。

4.2 算法空间复杂性分析

如前面 Delaunay 算法介绍所述, 分割-归并算法可以有带分割方法、网格分割方法和与本文采用四叉树分割方法 (如图 5 所示)。从图中可见, 因为

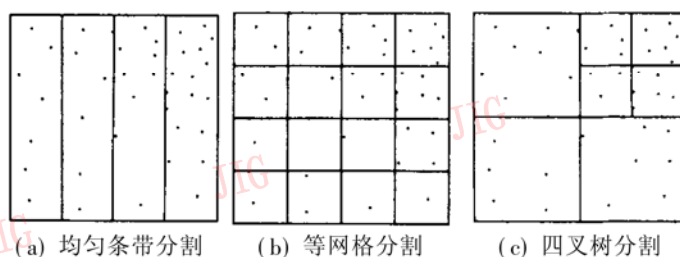


图 5 3 种不同的空间分割方法示意图

条带分割方法和网格分割方法不能够随着空间点分布的不同而产生变化,所以在空间点分布产生异常时,将会降低算法的效率,即算法不够稳定,虽然可以采用非均匀的条带分割和网格分割方法来在一定程度上适应空间点分布上的变化,但这样将会大大降低限定边和限定多边形嵌入的效率,而四叉树分割方法则会随着点的空间分布不同而不同,从而在空间点分布差别很大的情况下,仍然可以保证较稳定的算法执行效率,同时四叉树结构也非常有利于限定边和限定多边形的嵌入。

5 结 论

本文给出了一个基于四叉树结构的数字地表模型快速生成算法。实测结果表明,算法有着非常好的性能,而且也非常稳定。通过分析,算法的时间效率(复杂度)近似为 $O(n\log(n))$ 。通过算法的空间复杂性分析,可以看出算法可以适应不同的点空间分布情况,而且采用四叉树结构也非常有利于限定边和限定多边形的嵌入。

参 考 文 献

- 1 Fang T P, Piegl L A. Algorithm for delaunay triangulation and convex-hull computation using a sparse matrix [J]. *Computer Aided Design*, 1992, 24(8): 425~ 436.
- 2 De Floriani L, Puppo E. An on-line algorithm for constrained delaunay triangulation [J]. *Graphical Models and Image Processing*, 1992, 54(3): 290~ 300.
- 3 Piegl L A, Richard A M. Algorithm and data structure for triangulating multiply connected polygonal domains [J]. *Computers & Graphics*, 1993, 17(5): 563~ 574.
- 4 彭辉. 基于 MicroStation 的数字高程模型的建立及其应用研究 [D]. 武汉: 武汉测绘科技大学, 1998.

- 5 武晓波, 王世新, 肖春生. Delaunay 三角网生成算法研究 [J]. *测绘学报*, 1999, 28(1): 28~ 35.
- 6 王会成, 牟欣, 张利波. Trimmed 曲面的三角剖分算法研究 [J]. *计算机辅助设计与图形学学报*, 1997, 9(3): 218~ 222.
- 7 Dwyer R A. A faster divide-and-conquer algorithm for constructing delaunay triangulations [J]. *Algorithmica*, 1987, 2(2): 137~ 151.
- 8 Guibas L, Stolfi J. Primitives for the manipulation of general subdivisions and the computation of voronoi diagrams [J]. *ACM Transactions on Graphics*, 1985, 4(2): 75~ 123.
- 9 Clarkson K L, Shor P W. Applications of random sampling in computational geometry [J]. ii. *Discrete and Computational Geometry*, 1989, 4(1): 387~ 421.
- 10 Guibas L, Knuth D, Sharir M. Randomized incremental construction of delaunay and voronoi diagrams [J]. *Algorithmica*, 1992, 7(4): 381~ 413.
- 11 Dwyer R A. Higher-dimensional voronoi diagrams in linear expected time [J]. *Discrete & Computational Geometry*, 1991, 6(4): 343~ 367.
- 12 Maus A. Delaunay triangulation and the convex hull of n points in expected linear time [J]. *BIT*, 1984, 24(2): 151~ 163.



谢传节 1971年生,中国科学院地理科学与资源研究所资源与环境信息系统国家重点实验室博士生。主要研究方向为GIS系统软件设计,特别是多维GIS系统中的数据模型和算法研究。



万洪涛 1969年生,2000年获中科院地理所资源与环境信息系统国家重点实验室博士学位,现在中科院遥感所进行博士后研究。主要从事地理信息系统、遥感以及水文方面的研究。