

基于 Bowyer-Watson 三角网生成算法的研究

周雪梅¹, 黎应飞²

ZHOU Xuemei¹, LI Yingfei²

1. 贵州大学 计算机科学与信息学院, 贵阳 550025

2. 贵州大学 教务处, 贵阳 550025

1. Institute of Computer Science and Technology, Guizhou University, Guiyang 550025, China

2. Office of Academic Affairs, Guizhou University, Guiyang 550025, China

ZHOU Xuemei, LI Yingfei. Algorithm research to generate triangulation network based on Bowyer-Watson. Computer Engineering and Applications, 2013, 49(6): 198-200.

Abstract: TIN (Triangulated Irregular Network) has better performance on shaping terrain. The generation algorithm has been great concerned. This paper has discussed the data structure design of triangulation, and designed and implemented the algorithm based on Bowyer-Watson idea that is an incremental insertion algorithm. This paper analyzes why the algorithms may arise the phenomenon of cross during the experiment, and gives the improved idea. The improved algorithm has been used to visualize the terrain modeling, to obtain good results, for the research triangulation has some value.

Key words: Triangulated Irregular Network (TIN); Delaunay triangulation; convex hull

摘 要: 不规则三角网 (Triangulated Irregular Network, TIN) 在表示地形的形态方面具有较好的表现, 其生成算法一直备受关注。讨论了三角网的数据结构的设计, 采用逐点插入算法中的 Bowyer-Watson 算法思想为研究重点, 设计并实现了该算法, 对算法实验过程中可能出现的交叉现象进行分析, 给出算法的改进。该改进算法已用于地形的可视化建模中, 获得了较好的效果, 对于三角剖分的相关研究具有一定的价值。

关键词: 不规则三角网; Delaunay 三角剖分; 凸壳

文献标志码: A **中图分类号:** TP391 **doi:** 10.3778/j.issn.1002-8331.1108-0137

1 引言

在数字地形建模中, 不规则三角网 (TIN) 通过不规则分布的数据点生成的连续三角面来逼近地形表面。就表达信息的角度而言, TIN 模型的优点在于它能从不同层次的分辨率来描述地形表面。与格网数据模型相比, TIN 模型在某一特定分辨率下能以更少的空间和时间更精确地表示更加复杂的表面。特别是当地形包含有大量特征如断裂线、构造线时, TIN 模型能更好地顾及这些特征, 从而能更精确合理地表示地表形态^[1]。在生成 TIN 的各种算法中, 最常用的是 Delaunay 三角剖分法。这是因为在所有可能的三角网中, Delaunay 三角网在地形拟合方面的表现是最为出色的。如何快速、高效地构建 Delaunay 三角网, 一直是众多学者研究和关注的热点。迄今为止出现了不少成熟的算法, 如分割-合并算法、逐点插入法以及三角网生长法等。其中三角网生长算法由于算法效率较低, 目前较

少采用; 逐点插入法虽然实现较简单, 占用内存较小, 但其时间复杂度差, 效率较低; 分割-合并算法最为高效, 但相对复杂, 由于其深度递归, 对内存要求比较高。它们的平均时间复杂度分别有 $O(n)$ 、 $O(n \lg n)$ 和 $O(n^2)$ ^[2]。目前国内外多数的研究集中在逐点插入算法的搜索策略的改进^[3-4]和分治算法与逐点插入算法的合成^[5-6]两个方面。本文所研究的 Bowyer-Watson 算法就是逐点插入算法的一种。

2 数据结构的设计

一般三角网可以采用以下两种数据结构来表示: 一种是点-边-三角形 (即所谓的 Quad_Edge 结构); 另一种是点-三角形结构。很明显点-三角形结构所需的存储空间要小于前者; 但是明确边的数据结构的设计, 可以使与边有关的属性的存储和维护更为方便, 易于检索, 三角网间的拓扑关系更加明确。因此在本文中所使用的数据结构是点-

作者简介: 周雪梅 (1977—), 女, 讲师, 主要研究领域为图形图像处理; 黎应飞 (1973—), 男, 讲师, 主要研究领域为水利水电项目管理。

E-mail: xzmzhou163@163.com

收稿日期: 2011-08-12 **修回日期:** 2011-10-17 **文章编号:** 1002-8331(2013)06-0198-03

CNKI 出版日期: 2011-12-09 <http://www.cnki.net/kcms/detail/11.2127.TP.20111209.1002.051.html>

边-三角形的数据结构,其具体如下:

(1)离散点表:离散点表作为所有点坐标数据的数据库,建立其索引便于以后的边表和三角形表中点索引的查询。

```
class DiscretedPoint:public Object{
    float x,y; //离散点的坐标
    long index; //点的索引
}

class Edge:public Object{
    long Start; //边的起点
    long End; //边的终点
    long LeftTriangle; //边的左三角形索引
    long RightTriangle; //边的右三角形索引
    long index; //边的索引
}
```

(2)有序边表:记录三角网生成期间所使用过的边,以及最后存储的三角形间共享边信息。在有序边表中,规定边的起点的索引始终小于边的终点的索引。

```
class Edge:public Object{
    long Start; //边的起点
    long End; //边的终点
    long LeftTriangle; //边的左三角形索引
    long RightTriangle; //边的右三角形索引
    long index; //边的索引
}
```

(3)三角形网表:记录最终生成的 TIN 数据模型中的三角网之间的拓扑关系。该数据结构是以经典的 LTL(Lawson's Triangle List)表结构来存放三角网信息的。

```
class TriangleNet:public Object{
    long NodeA; //三角形的顶点 A 的坐标索引
    long NodeB; //三角形的顶点 B 的坐标索引
    long NodeC; //三角形的顶点 C 的坐标索引
    long AdjA; //顶点 A 的对边相邻的三角形
    long AdjB; //顶点 B 的对边相邻的三角形
    long AdjC; //顶点 C 的对边相邻的三角形
    long index; //三角形的索引
}
```

3 TIN 的生成算法的研究

3.1 Bowyer-Watson 算法的基本思想

Bowyer-Watson 算法的基本思想如下^[7]: (1)假定已生成了连接若干个顶点的 Delaunay 三角网格; (2)加入一个新的节点,找出所有外接圆包含新加入节点的三角形,并将这些三角形删除,形成一个空腔; (3)空腔的节点与新加入的节点连接,形成新的 Delaunay 三角形网格; (4)调整数据结构,新生成的三角形的数据填充被删除三角形的数据,余者添加在数组的尾部; (5)返回第(2)步,直至所有的节点都加入为止。

Bowyer-Watson 算法属于逐点插入法的一种,其前提条件是需要生成一个 Delaunay 三角网的初始网格。考虑算法的效率,选用的是格雷厄姆算法所生成的一个包含所有的离散数据点的凸壳作为初始网格,该算法的复杂度为 $O(n \lg n)$ 。

```
根据上述 Bowyer-Watson 算法的基本思想,实现如下:
void Bowyer_Watson_Algorithm(){
    初始三角形数组
    将所有待加入的离散数据点放入点堆栈中
    while(点堆栈不为空){
        初始化边列表为空;
```

```
点堆栈出栈,即弹出一个待加数据点;
for(i=0;i<三角形数组中元素的个数;i++){
    if(待加数据点在三角形的外接圆内){
        将该三角形的三条边与边列表中的所有边进行比较;{
            if(该边已经在边列表中存在)
                边列表中的这条边的权值变为 2;
            else if(该边不在边列表中)
                则将该边加入边列表;
        }
        将该三角形从三角形数组中移出;
    }
    for(i=0;<边列表的长度;i++){
        if(边的权值为 1){
            将边与待加数据点形成的所有三角
            形加入到三角形数组中;
        }
    }
}
```

3.2 算法的改进

根据上述算法设计,经测试,在数据分布较为均匀的情况下,上述算法能较好地生成 Delaunay 三角网格,但是如果数据中存在有导致狭长三角形出现的离散数据点时,使用该算法就有可能使生成的三角网出现交叉现象(如图 1)。

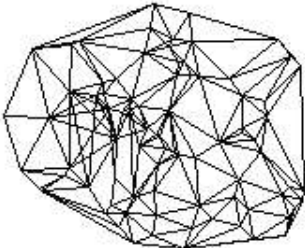


图1 生成的三角网中出现交叉现象

经分析,出现上述交叉现象,是由于在对新点所影响的全部三角形进行遍历,当出现图 2(a)中所示情况时,就会导致交叉现象的发生。这主要是因为,当受新加入节点影响三角形被删除时,形成的空腔不只一个,这是导致生成的三角网中出现交叉现象的直接原因。



图2 交叉现象出现的原因

从图 2(a)中,可以看到新加入点影响的三角形中间有一个未受影响的三角形存在,因此使受影响的三角形出现分离,形成了两个空腔(见图 2(b)),致使交叉现象发生(图 2(c))。

由于交叉现象的出现将影响整个三角网的生成质量,因此需要将受影响的三角形限定在一个空腔内,且新插入点在该空腔内,所以插入一个新节点时,首先要判断它在哪个三角形内(或上),再从该三角形开始向其邻接三角形进行查询,找出相邻三角形中包含新节点的三角形,形成

空腔。这样做保证了新点一定在所构成的空腔内,且该空腔是惟一的。具体如下:

(1)找出新加入节点所在的三角形。

(2)将该三角形的三条边的索引,节点信息以及其所在三角形的邻接三角形的信息压入邻接边栈,并修改邻接边的权值。

(3)如果邻接边栈为空,则直接跳到第四步;否则,判断栈顶的边信息,如果其权值为2(即为共享边),则直接弹出;如果其邻接三角形的外接圆不包含该新节点,则将该邻接边信息压入空腔边栈。如果其邻接三角形的外接圆包含新加入节点,则修改其权值为2,并将受影响三角形的索引压入受影响三角形列表;如果包含新加入节点,则继续将受影响三角形的邻接三角形的信息压入邻接边栈,直到邻接边栈为空,则已形成空腔。

(4)从空腔边栈中取出一条边,让其与新加入节点构成新的三角形,如果受影响三角形列表不为空,则取出一索引号,修改索引所指的受影响三角形的信息;如果受影响三角形列表为空,则将新生成的三角形加到三角形列表的尾部,直到空腔边栈为空。

(5)继续加入下一新节点,重复上述过程,直到所有的新节点都插入完毕。

4 实例测试

在后续的实验中,针对图3(a)所示的一组等高线,对该组等高线采样得到相应离散数据点(791个),根据采样的离散点生成的TIN模型。实验证明,Bowyer-Watson的改进算法与原始Bowyer-Watson算法,在没有狭长三角形存在的情况下,这两种算法的效率相当,生成的三角网也一样;当有狭长三角形存在,后者会出现交叉现象的情况,前者仍能较好地生成三角网格。从图3(b)中最初生成的TIN模型的三角分布来看,等高线密集的地方,构成三角形网格

的三角形也相对小而密;等高线稀疏的地方,三角网的分布相对大而稀。但是却有许多地方与事实不符,这主要是由于在TIN模型的生成中尚未考虑“平三角形”以及地性线的情况。在图3(c)中,将三角网中生成的平三角形予以消除,从图中可以看出地形的大致特征,但是由于未对地性线进行处理,导致生成的TIN模型不够完善。

5 结束语

本文研究和分析了以Bowyer-Watson算法为基础的不规则三角网生成算法,该算法能较好地建立三角网。该算法在点插入过程中,会对已建立的三角网进行遍历检索,以查找因该点插入时而受影响的三角形,这个过程效率由三角网的数目和三角形所在的位置决定。在实验中,发现在凸壳生成算法中,离散点已按角度和距离进行排序,对于新加入节点,其在数组中的位置与其在空间中的位置邻近的点相邻近;这时新插入一个节点,先在上一节点插入时所更新和增加的三角形数组内查找新节点所在三角形的命中率将会提高,这样做不仅可帮助解决交叉现象,同时也给算法的效率带来一定的提高。此外,由于TIN模型采用的是点-边-三角形的数据结构,在算法的实现中会多次对边进行遍历,随着三角形数目的增多,新点所在三角形有可能会在很靠后的位置,采用边队列的尾部遍历的效率较高。本文只对单一分块的数据进行算法设计,在实际应用中,所面临的离散点将会是巨大的,如同时考虑加入分治算法的思想将会大大提高算法的效率,此不是本文的重点,在此不再赘述。

参考文献:

- [1] 李志林,朱庆.数字高程模型[M].武昌:武汉大学出版社,2001:29-59.
- [2] 蒋瑜,杜斌,卢军,等.基于Delaunay三角网的等值线绘制算法[J].计算机应用研究,2010,27(1):101-103.
- [3] 宋占峰,蒲浩,詹振炎.基于三角网数字地面模型快速定位算法的研究[J].中国铁道科学,2002,23(1):63-66.
- [4] Zhou Sheng, Jones C B. HCPO: an efficient insertion order for incremental Delaunay triangulation[J]. Information Processing Letters, 2005, 93(1): 37-42.
- [5] 向传杰,朱玉文.一种高效的Delaunay三角网合并生成技术[J].计算机应用,2002,22(11):34-36.
- [6] 吴宇晓,张登荣.生成Delaunay三角网的快速合成算法[J].浙江大学学报,2004,31(3):343-348.
- [7] 郭吉明,沈隆均,张景琳.Delaunay三角网格的一种快速生成法[J].数值计算与计算机应用,2001,22(4):267-275.
- [8] 周培德.计算几何——算法分析与设计[M].北京:清华大学出版社,2000.

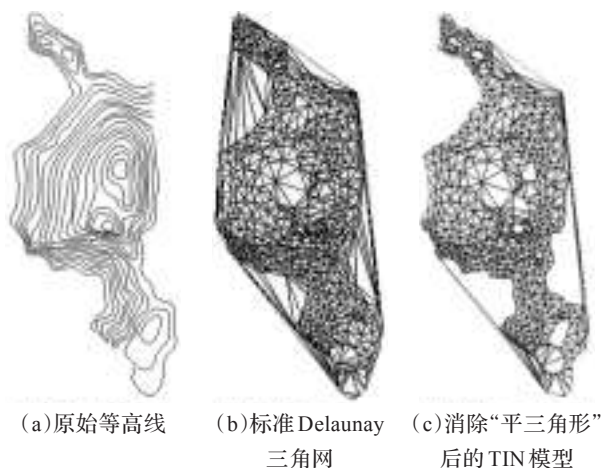


图3 TIN模型的生成