

## 一种基于动态正方形的 TIN 的构建算法\*

彭 李, 刘少华, 卢学军

(长江大学 地球科学学院, 湖北 荆州 434023)

**摘要** 为了提高不规则三角网的构网效率, 提出了一种改进算法, 该算法执行效率高, 构网速度快, 实现较简单, 并用 VC++6.0 编程语言对算法进行实现, 最后用实验数据对算法进行了测试。

**关键词** 不规则三角网 Delaunay 三角网 算法 动态正方形

**中图分类号**: P224.2 **文献标识码**: A **文章编号**: 1672-4097(2007)02-0011-02

## 1 引言

数字高程模型(DEM)主要用于描述地面起伏的状况, 可用于提取各种地形参数, 如坡度、坡向、粗糙度等, 并进行通视分析、流域结构生成等应用分析<sup>[1]</sup>。目前, DEM模型的建立和应用已经成为GIS领域的核心, 得到了广泛使用。不规则三角网(TIN)是一种表示数字高程模型(DEM)的方法, 它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, 既减少规则格网方法带来的数据冗余, 同时在计算(如坡度)效率方面又优于纯粹基于等高线的方法。因此它是一种有效表示数字高程模型的方法。TIN的构建算法已经成为GIS研究的一个热点问题, 诸多专家和学者提出了相应的改进算法, 虽然算法速度比较快, 但其实现方法相对较难, 针对这种情况, 本文提出一种基于动态正方形的方式来限定目标点的搜索范围, 该算法思想简单, 实现容易, 构网效率也比较高。

## 2 TIN的构建算法及改进

## 2.1 构建算法

Delaunay三角网是目前公认的最优三角网, 它满足如下法则: Delaunay三角网为相互邻接且互不重叠的三角形的集合, 每一个三角形的外接圆内不包含其它点。Delaunay三角网由对应Voronoi多边形共边的点连接而成。Delaunay三角形由三个相邻点连接而成, 这三个相邻顶点对应的Voronoi多边形有一个公共的顶点, 此顶点同时也是Delaunay三角形外接圆的圆心。Delaunay法则就是在使用所采集的离散点组成三角网时, 确保每个三角形的内角都是锐角或者三边的长度近似相等, 避免出

现过大的钝角和过小的锐角<sup>[2]</sup>。

三角网生长算法的基本步骤: 首先从所采集的离散点集V中选择任意点 $P_1$ 作为起始点, 查找距离此点最近的点 $P_2$ , 然后连接 $P_1P_2$ 作为基线; 在基线的右边应用Delaunay法则搜寻第三点生成Delaunay三角形, 以三角形的两条新边(从基线起始点到第三点及第三点到基线终止点)作为新的基线; 重复该过程直至所有的基线处理完毕。

## 2.2 算法改进

以上算法大部分的时间都是在大量离散数据点中搜寻给定基线符合要求的邻域点, 而且每次找点都要遍历整个离散数据点, 极大地降低了构网的速度。为了减少搜索时间, 本文提出一种以动态正方形的方式限定目标点的搜索范围。首先计算扩展基边的中点坐标, 以该点为中心作一个正方形(如图1(a)), 该正方形的大小根据参与构网的点数及点的密度动态确定, 使得落在正方形区域内的点数为50个(经验值), 将搜索范围定位在该正方形内, 在搜寻第三点时, 它应满足如下要求:

① 在直线的右边。判别点在直线右边的直线方程为  $F(x, y) = y - Ax - B$ 。

其中  $A = (y_2 - y_1) / (x_2 - x_1)$ ,  $B = (x_2 * y_1 - x_1 * y_2) / (x_2 - x_1)$ ,  $x_1, y_1, x_2, y_2$  分别为扩展边两端点的坐标,  $F(x, y) > 0$  表示离散数据点位于直线的左边,  $F(x, y) < 0$  表示离散数据点位于直线的右边,  $F(x, y) = 0$  表示离散数据点位于直线上。

②  $\theta$ 角最大原则。根据余弦定理  $c^2 = a^2 + b^2 - 2ab\cos(\theta)$  可确定, 其中 $\theta$ 角是目标点与扩展基边两顶点组成的夹角, 为了减少计算量, 没必要计算出 $\theta$ 角, 只要算出余弦值即可。同时在Edge边数据结构中加入了边的使用次数, 这样就没有必要对每个

\* 江西省数字国土重点实验室开发研究基金资助(DLLJ200501); 湖北省高等学校优秀中青年团队计划项目基金资助(T200602); 长江大学发展基金资助(2004Z0115)

新生成的三角形的两条新边都向外扩展生长,有效地减少了扩展的边个数,从而提高了构网速度,如果在正方形内没有找到满足条件的点,则需要动态地将正方形的边长扩大(如图1(b)),直至找到满足条件的点为止或其右侧无点。

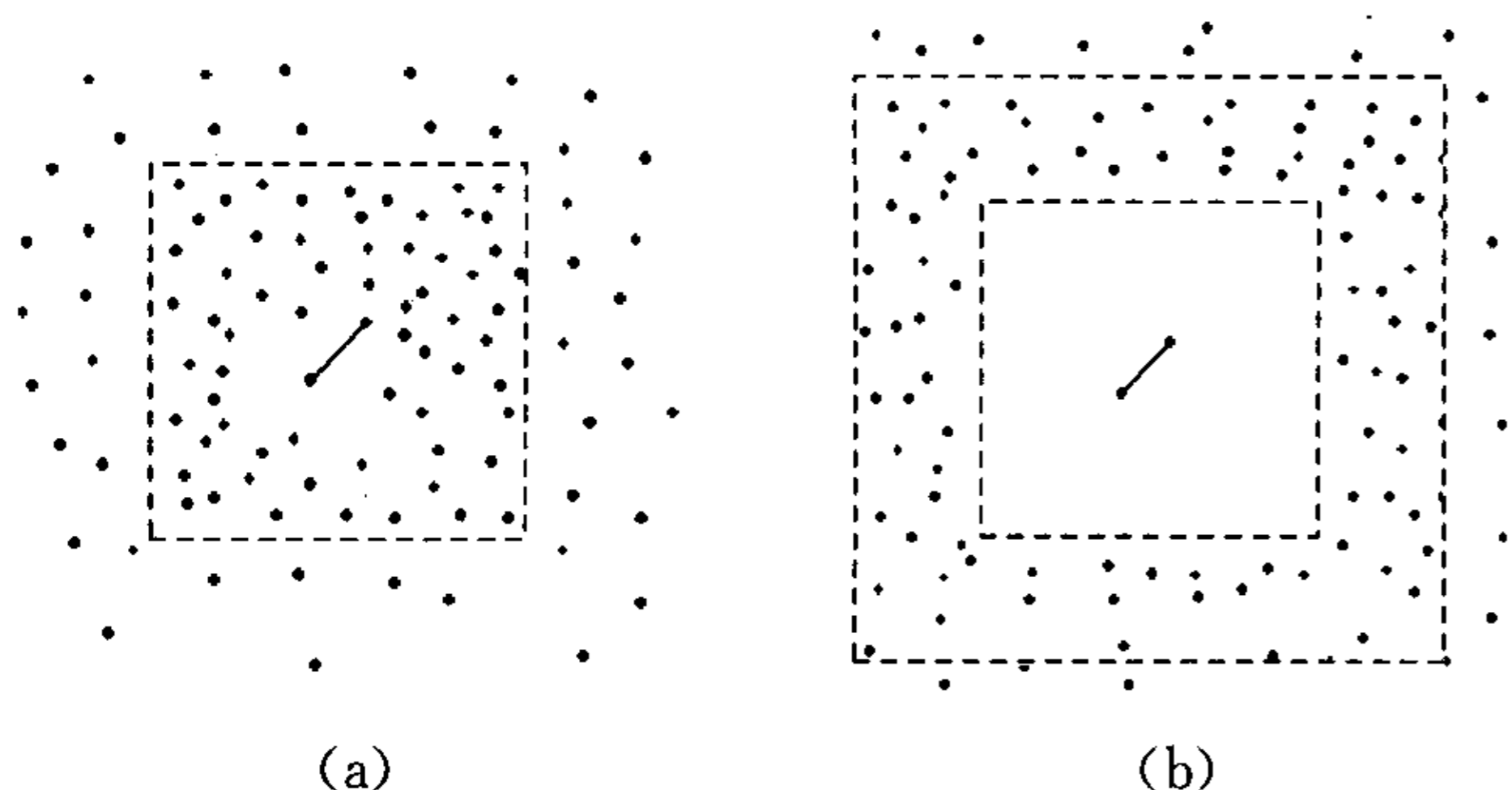


图1 搜索范围的确定

### 3 算法的数据结构

数据结构定义如下:

```

struct Point //点结构
{
float x, y, z; //离散点的坐标
int nPntID; //点号
};
struct Edge //边结构
{
int P[2]; //边的顶点号
int nUsedTimes; //记录边使用的次数
int nEdgeID; //边号
};
struct Triangle //三角形结构
{
int P[3]; //三角形的顶点号
int E[3]; //三角形的边号
int T[3]; //三角形的相邻三角形号
};

```

### 4 具体实现步骤

本算法的详细步骤如下:

- 步1 从原始数据文件中读入数据点集  $V_i (i = 1, 2, 3 \dots n)$ , 并储存于点表中。
- 步2 在点集  $V_i$  中任取一点  $P_1$ , 以点  $P_1$  为基点寻找与它最近的一点  $P_2$ 。连接  $P_1P_2$ , 就得到了

三角形的一条基边, 将其加入到边链表中, 并以该边为扩展基边, 转步3。

- 步3 计算出扩展基边的中点坐标, 以该点为中心作一个正方形, 在该正方形内搜寻一点, 使得该点在扩展边的右边且与扩展基边两端点连成直线组成的夹角为最大。如果点存在, 就构成一个三角形, 将所有新生成的边与三角形信息用相应的链表存储, 边每使用一次, 其数据结构中的  $nUsedTimes$  就加1; 否则, 把正方形扩大, 直至找到满足条件的点或右边无点。转步4。
- 步4 在边链表中取出下一条边, 以该边为扩展基边向外进行扩展。如果该边的使用次数为2或是右边没有点, 该边不进行扩展; 否则, 转步3进行扩展, 同时存储新生成的边和三角形。转步5。
- 步5 重复步4, 直至边链表中的所有边都进行了扩展, 就结束构网。

### 5 算法实验及结论

笔者在 VC++ 下实现了上述算法, 并用实验数据进行了验证, 本算法生成的三角网模型表明此算法较稳定, 构网速度快, 生成的三角网是 Delaunay 三角网。本算法与原始生长算法相比, 三角网构建的速度有明显提高, 前者比后者的速度快几倍, 甚至几十倍, 这与参与构网点数有关, 点越多前者的速度越明显。原始生长算法建立 Delaunay 三角网虽然原理容易理解, 算法的实现也比较简单, 但大部分的时间都是在大量离散数据点中搜寻给定基线符合要求的邻域点, 而且每次找点都要遍历整个离散数据点, 极大地降低了构网的速度。通过以上改进后大大提高了 Delaunay 三角网生成速度, 提高了构网的效率。

#### 参考文献

- 1 邬伦, 刘瑜, 张晶等. 地理信息系统——原理、方法和应用 [M]. 北京: 科学出版社, 2001
- 2 李志林, 朱庆. 数字高程模型 [M]. 武汉: 武汉大学出版社, 2001

## An Algorithm of Constructing TIN Based on Dynamic Square

Li Peng, Liu Shaohua, Lu Xuejun

(Geoscience institute, Yangtze University, HuBei, JingZhou, 434023)

**Abstract** In order to improve the efficiency of constructing TIN, the papers proposes a kind of improved algorithm, which carries out efficiently, constructs quickly, and realizes simply, and achieves the algorithm in VC++6.0 programming language, finally uses the empirical data to test the algorithm.

**Key words** TIN; Delaunay; Algorithm; Dynamic square