

一种基于图的平面点集 Delaunay 三角剖分算法*

马小虎 董军 潘志庚 石教英

(浙江大学 CAD&CG 国家重点实验室 杭州 310027)

摘要 本文提出了一种基于图的平面点集 Delaunay 三角剖分算法。该算法首先求出平面点集的欧几里得最小生成树,然后逐次加入一边构造三角形网格,最后按最小内角最大的三角化准则,通过局部变换,得到平面点集的 Delaunay 三角剖分。本文同时阐述了它的对偶图:平面点集的 Voronoi 图的概念和性质。

关键词 欧几里得最小生成树,平面点集, Delaunay 三角剖分, Voronoi 图

1 引言

在 CAD/CAM、计算机图形学、有限元分析等领域中,经常要处理这样的问题:给定一组散乱数据 $P_i = \{x_i, y_i, z_i\} (i=1, 2, 3, \dots, n)$, 首先要在 XOY 平面上对其进行三角剖分,然后在三角域上构造插值曲面,最后以插值曲面为基础进行各种分析、计算及其它处理。因此平面点集的三角剖分是一个值得研究的基本问题^[1]。

目前,比较常用的二维三角剖分的优化标准有最大-最小距离标准、圆标准、最大-最小角标准、最大-最小高标准、Thiessen 标准等^[1]。这些标准的一个基本出发点就是尽量避免平面三角形出现太尖的情况。因为根据逼近论的分析结果,三角曲面的逼近误差与三角区域的最小内角有关,避免过尖的三角形就可提高逼近精度,这一问题在有限元分析中显得尤为重要。

Delaunay 三角剖分具有最小内角最大以及平均形态比最大的性质,因此它是给定区域的点集的最佳三角剖分^[2]。本文提出一种平面点集的 Delaunay 三角剖分算法,是作者在研究表面重构及三角形网格简化中得到的结果,并在其中得到了较好的应用。该算法的基本思想可用如下步骤来描述:①求出平面点集的欧几里得最小生成树;②逐次加入一

条边,构成三角形网格,直到最后一条边加入为止;③按最大-最小内角准则,通过局部变换,得到 Delaunay 三角剖分。

下面介绍平面 Delaunay 三角剖分的基本性质,详细描述本文提出的算法,及其采用的主要数据结构并给出一些实例。

2 平面点集 Delaunay 三角剖分的基本性质

令 $P = \{P_1, P_2, \dots, P_N\}$ 为平面域(R^2)上 N 个离散点的集合,尽管有多种方法实现点集 P 的三角剖分,但是俄国数学家 Delaunay 在 1934 年证明:必定存在且仅存在一种剖分(一般称之为 Delaunay 三角剖分)算法,使得所有三角形的最小内角之和最大。因此, Delaunay 三角剖分能够尽可能地避免病态三角形的出现,而这正是有限元分析所期望的,因此很适合于有限元网格自动生成。

一般情况下,一个完整的 Voronoi 图应该由多个 Voronoi 多边形组成,第 i 个 Voronoi 多边形的数学表达形式如下:

$$V_i = \{x \in R^2: \|x - P_i\| \leq \|x - P_j\|, j = 1, 2, \dots, N, j \neq i\} \quad (1)$$

式(1)中 $i=1, 2, \dots, N$, $\|x - P_i\|$ 表示平面域上点 x 和节点 P_i 之间的欧氏距离,从(1)式可知,

Voronoi 多边形 V_i 内任意点 x 到节点 P_i 的距离比到点集 P 中任何其它节点的距离更近,因此 V_i 由节点 P_i 和每个相邻节点的垂直平分线所形成的开式半平面的交集组成,故 V_i 必为凸多边形。一般情况下,Voronoi 图的一个顶点同时属于 3 个 Voronoi 多边形,每个 Voronoi 多边形内有且仅有一个节点。连接 3 个共点的 Voronoi 多边形分别对应的每个节点则形成一个 Delaunay 三角形,所有这样的三角形的集合就是著名的 Delaunay 三角剖分^[3]。图 1 给出了 16 个节点的 Delaunay 三角剖分及其对偶的 Voronoi 图。

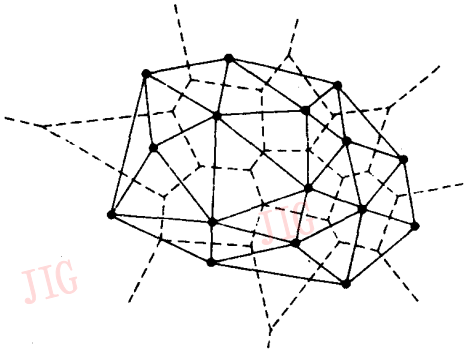


图 1 Delaunay 三角剖分(实线)和 Voronoi 图(虚线)
Fig. 1 Delaunay triangulation (solid line)
and Voronoi diagram (dotted line)

通常都是通过计算 Voronoi 图得到对偶的 Delaunay 三角剖分。尽管 Delaunay 三角剖分与 Voronoi 图互为对偶,但是 Delaunay 三角剖分的应用范围比 Voronoi 图要大得多。Delaunay 三角剖分的一个重要性质就是所谓的“空圆盘”性质,对于任意 Delaunay 三角形,其开式圆盘(即该三角形外接圆的内部区域)不包含其它任何节点,所有 Delaunay 三角形互不重叠,且完整地覆盖整个问题域,近年来出现了许多直接计算 Delaunay 三角剖分的算法^[4]。本文提出的方法是先求出平面点集的欧几里得最小生成树,逐步加入边构成三角形网格,再通过局部变换,得到平面点集的 Delaunay 三角剖分。

3 基于图的 Delaunay 三角剖分算法

3.1 基本概念

3.1.1 定义

定义 1: 设 $P = \{P_1, P_2, \dots, P_n\}$ 是一个平面点集,则欧几里得最小生成树(EMST)是一棵连接 P 中所有点的树,并且其边长之和为最小,简称最小生成树。

定义 2: 设图 $G = (V, E)$, 其中 V 是顶点集, E 是边集,若有边 $e_{ik} = (v_i, v_k)$, $e_{ij} = (v_i, v_j)$ 则称 e_{ik} 与 e_{ij} 为一对相关边。有向边 $v_i v_k$ 与 $v_i v_j$ 之间的夹角为相关边对 e_{ik}, e_{ij} 的夹角。

定义 3: 由 EMST 或当前图中相关边对组成的表称为相关边对表(简记为 IEPL)。

3.1.2 约束

在向 EMST 中加入边形成三角形网络时,必须满足如下约束条件:

约束 1: 三角形相互之间是不相交的,即两个三角形除端点外无交点,称此约束为三角形相交约束。

约束 2: 三角形相互之间是互不包含的,即任一三角形不完全包含其它三角形,称此约束为三角形包含约束。

3.2 三角剖分算法

基于图的平面点集的 Delaunay 三角剖分过程可用下面的算法来描述:

算法 1 /* 平面点集的 Delaunay 三角剖分 */

A1. 求出平面点集 $P = \{P_1, P_2, \dots, P_n\}$ 的最小生成树 EMST(P)(参见算法 2)。

A2. 根据 EMST(P) 求出初始相关边对表 IEPL(P)。

A3. 根据相关边对的夹角,从小到大排序相关边对表 IEPL(P)中的所有相关边对。

A4. 初始化三角形网格表 TML(triangle mesh list)^[5]。

A5. 当 IEPL(P)中还存在一个边对没有处理时,则从中选择夹角最小的相关边 e_{ij}, e_{ik} 对,并置 mark 标志为 1,如果该相关边对构成的三角形 $\Delta v_i v_j v_k$,同时满足约束 1 和约束 2,那么做以下操作:①把三角形 $\Delta v_i v_j v_k$ 加入到 TML 中;②把新边 (v_j, v_k) 加入到 EMST(P)中;③把新边 (v_j, v_k) 引入的新的相关边对,根据其夹角插入到排序的 IEPL(P)中。

A6. 按最大-最小内角准则,通过局部优化变换,得到 Delaunay 三角剖分(参见算法 3)。

算法 2 /* 求 EMST(P) */

B1. 计算出点集 P 中所有两点之间的距离,并存入距离数组 Dist[M]。其中 $M = n(n-1)/2$, n 为点集 P 中点的个数,数组 Dist 中每一元素包含三项;Start 存放起点, end 存放终点, length 存放两点之间的距离。

B2. 按数组 Dist 中元素的 length 项的值从小到大排序数组 Dist。

B3. $k \leftarrow 1, e \leftarrow 0$, 并置 $T(P)$ 为空。

B4. 当 $e \leq n-1$, 则做以下操作:

取 $u = \text{Dist}[k]. \text{start}, v = \text{Dist}[k]. \text{end}$, 置 $k \leftarrow k + 1$

若边 (u, v) 与 $T(P)$ 中的边不构成回路, 则把边 (u, v) 加入到 $T(P)$ 中, 并置 $e \leftarrow e + 1$

算法3 /* 局部优化 */

C1. 从 TML 表形成三角形网格的边表 EL。

C2. 对边表 EL 中的每一条非边界边, 做步骤

C3。

C3. 取出以此边为邻接条件的两个相邻三角形, 构成一个四边形。如果这个四边形是凹的, 则不做任何处理。如果这个四边形是凸的, 则计算两个三角形的最小内角 α_1 , 交换对角线计算出两个新的三角形的最小内角 α_2 , 如果 $\alpha_2 > \alpha_1$, 则以交换对角线所得到的两个新三角形替换原来的两个三角形, 并按新的三角形修改 TML, 交换边计数变量 interchaged-edge 加 1(初始值为 0)。

C4. 如果 interchaged-edge = 0, 则结束, 否则, 转步骤 C1。

3.3 算法的时间复杂度

由于算法 1 的第一步 A1 即为算法 2, 所以, 为了估算算法 1 的时间复杂度, 首先要估算出算法 2 的时间复杂度。而算法 2 中最耗时间的步骤是 B1, B2 和 B4, 其中步骤 B1 需 $O(n^2)$ 的时间, 步骤 B2 和 B4 所需的时间都不超过 $O(n^2)$, 因此, 算法 2 的时间复杂度为 $O(n^2)$, 其中 n 为点集 P 中的点的个数。

另外, 算法 1 中除第一步 A1 外, 耗时较多的为步骤 A5 和 A6, 步骤 A6(即为算法 3)所需的时间不超过 $O(n^2)$, 步骤 A5 所需的时间下限是 $O(n^2)$ 。因此算法 1 所需时间其量级也是 $O(n^2)$ 。

3.4 数据结构及实例结果

本算法已在 SUN SPARC 工作站上用 C 语言实现, 并已用在多细节层次模型构造的研究中。算法主要采用以下数据结构。

(1) 顶点表的结点类型

```
struct point {
float x; /* x 坐标 */
float y; /* y 坐标 */
}
```

(2) EMST 的结点类型

```
struct emst-node {
int v; /* 边的另一个顶点序号 */
struct emst-node * next;
}
```

(3) 相关边对表的结点类型

```
struct edgepair-node {
int j, k; /* 相关边对的另两个顶点序号 */
float alpha; /* 相关边对的夹角 */
int mark; /* 是否已被处理的标志 */
struct edgepair-node * link;
}
```

(4) 三角形表 TML 的结点类型

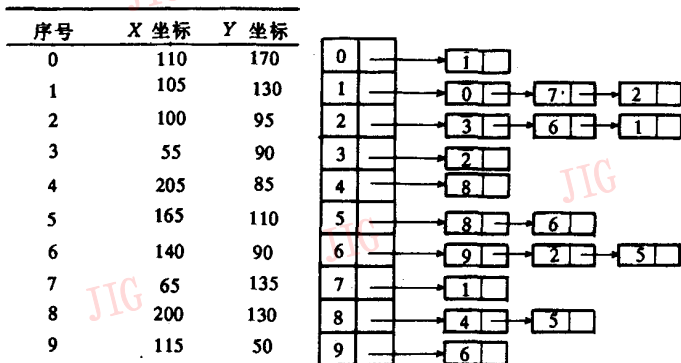
```
struct tri-node {
int vt [3]; /* 存放三角形的三个顶点序号 */
int nt [3]; /* 存放相邻三角形的三个顶点重复 */
}
```

注: 若为边界边, 则相邻三角形的分量为 $-1 * /$

(5) 三角形边表的结点类型

```
struct tri-edge-node {
int start; /* 边的起始顶点序号 */
int end; /* 边的终止顶点序号 */
int left; /* 边的左三角形序号 */
int right; /* 边的右三角形序号 */
}
```

图 2~4 为三角剖分的实例。

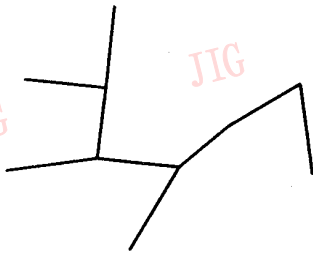


(a) 顶点数据表

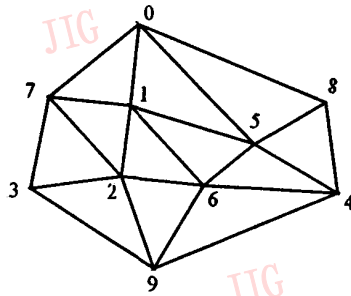
(b) EMST 链表

图 2

Fig. 2 (a) Vereex data table, (b) EMST link table



(a)最小生成树



(b)Delaunay 三角剖分

图 3

Fig. 3 (a)Minimum spanning tree, (b)Delaunay triangulation

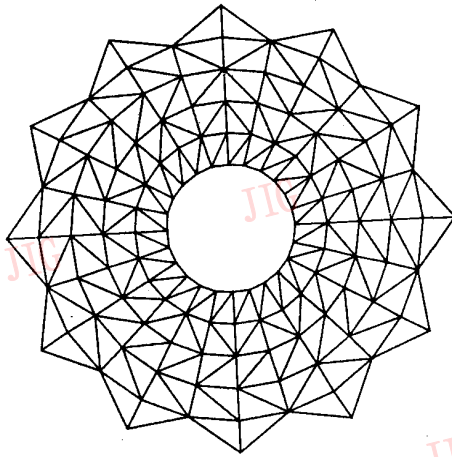


图 4 任意多边形区域内点集的 Delaunay 三角剖分

Fig. 4 Delaunay triangulation of point set in arbitrary polygon

4 结 论

(1)本文给出的算法,基于最小生成树,思路比较清晰,程序编写也比较简单,通用性强。

(2)只要对以上算法略加修改,就可用于任意多边形区域内点集的 Delaunay 三角剖分,图 4 给出了这样一个例子。

(3)把本文提到的算法思想推广到三维空间,在最小生成树的基础上生成曲面描述图,然后,对曲面描述图进行局部三角剖分,这样可以从一组空间散乱数据点重构物体的表面^[6]。

参考文献

- 1 姜寺山等. 用空间形状优化标准完成散乱数据的三角剖分. 计算机辅助设计与图形学学报, 1995, 7(4).
- 2 闵卫东, 唐泽圣. 二维任意域内点集的 Delaunay 三角划分的研究. 计算机学报, 1995, 18(5)
- 3 Preparate, F. P. and Shamos, M. I. Computational Geometry. Springer-Verleg, New York, 1985.
- 4 柯映林. 实现平面上散乱点自动三角划分的一种算法. 计算机工程, 1992, 18(2).
- 5 Lawson, C. L. . A software for C^1 surface interpolation. Mathematical Software III, New York. Academic press, 1977.
- 6 Robert Mencl. A Graph-based Approach to Surface Reconstruction. computer graphics forum, 1995, 14(3).



马小虎, 讲师。1991年毕业于南京大学计算机科学系, 获硕士学位。现在浙江大学CAD&CG国家重点实验室攻读博士学位。主要研究领域为计算机图形学、多媒体和虚拟现实。

A Graph-Based Algorithm for Generating the Delaunay Triangulation of a Planar Point Set

Ma Xiaohu, Dong Jun, Pan Zhigeng, Shi Jiaoying

(State Key Lab. of CAD&CG, Zhejiang University, HangZhou, 310027)

Abstract A graph-based algorithm for generating the Delaunay triangulation of a planar point set is presented in this paper. The first step is to calculate the Euclidean Minimum Spanning Tree (EMST) of the given points. The EMST is then augmented to triangle mesh. Finally the Delaunay Triangulation of the given points is obtained by local transformation according to Max-Min angle criteria. The concepts and properties of Voronoi diagram of a planar point set are also discussed in the paper.

Keywords Euclidean minimum spanning tree, Planar point set, Delaunay triangulation, Voronoi diagram

本刊采用“图象”等专业用词的说明

——答读者问

在历史和传统文化习惯上,通常描述自然性景物和科学技术领域的研究对象或实验结果时,用到“图象”一词,往往由于所涉内容的自然性,一般性和抽象含义而采用不加立人傍“亻”的“象”字,例如物理图象,数学图象等。尤其涉及到图象图形科学技术领域,许多专有名词和研究术语均采用“图象”而非“图像”,例如专业书籍《数字图象处理》(李叔梁等译,1981年)、《数字图象处理》(余英林等译,1982年)、《数字图象处理概论》(王宝兴等译,1986年)、《数字图象处理导论》(赵荣椿编著,1995年)和专有名词“中国图象图形学会”、《中国图象图形学报》等。

在英文词汇里,与“图象”对应的往往以 Image 居多,例如“Digital Image Processing”(《数字图象处理》,W. K. Pralt, 1978)、“Computer Image Processing and Recognition”(《计算机图象处理与识别》,E. L. Hall, 1979)。Image 本身含有自然性、一般性、抽象与泛指的意思,其后不加 S;若加 S 则为 Images,便带有具体、特定的含意。专家们翻译这些文字时是经过推敲的。

在祖国文字规范化方面,该词几十年来并非贯一,一时为“图象”,一时为“图像”。例如商务印书馆 1979 年出版的《辞源》,是继集 1915 年正编初版、1931 年续编初版、1939 年合订第 1 版、1979 年修订第一版的修订本,只有“图象”词条而无“图像”。其 1980 年出版的《现代汉语词典》亦为“图象”。到了 1996 年,同为商务印书馆出版,中国社会科学院语言研究所词典编辑室编辑的《现代汉语词典》中的图象词条已改变为“图像”。我们认为,在自然科学领域中,术语用词希望保持其固有的一致性与一贯性。为此我们电话请教了 1996 年版《现代汉语词典》编辑室,该室同志答称:在编该词典时,语言文字工作者倾向于用“像”字,而科技工作者认为“象”字更合适,意见并不完全统一。另外国家语言文字工作委员会并未对“图象”(像)一词的用字作最后审定。因此目前涉及图象的有关词汇用“象”还是“像”都不算错,都不能认为是非规范的字。鉴于以上情况,中国图象图形学报编辑出版委员会敬告广大读者:如果来稿作者不做特别说明,本刊编辑部在稿件编辑录排时,有关“图象”、“象素”、“遥感影象”等专业术语一律不用“像”,特此说明。