

三维散乱点集的曲面三角剖分

张永春 达飞鹏 宋文忠

(东南大学自动化研究所, 南京 210096)

摘 要 根据三维散乱点集构造曲面剖分在CAGD/CAD、反求工程等方面有着十分广泛的应用. 本文回顾了三维散乱点集曲面三角剖分的两种主要方法即平面投影法和直接剖分法, 对几种常用的算法作了较为详细的描述, 同时对剖分算法中经常出现的数据结构作了详细的介绍. 由于剖分采用的优化准则决定了剖分结果, 因此本文讨论了几种常用的剖分优化准则, 并举例对几种优化准则进行了较详细的分析比较. 最后简要地讨论了算法复杂度以及目前剖分的主要研究方向, 并指出在实际工程应用中, 对复杂曲面采样得到的散乱点集, 要得到光顺和保形的三角剖分, 需要新的剖分准则和算法.

关键词 图象处理(510·4050) 三维散乱点集 曲面三角剖分 数据结构 优化准则

中图分类号: TP391.72 **文献标识码**: A **文章编号**: 1006-8961(2003)12-1379-10

Surface Triangulations Based on 3D Arbitrary Point-sets

ZHANG Yong-chun, DA Fei-peng, SONG Wen-zhong

(Research Institute of Automation, Southeast University, Nanjing 210096)

Abstract Surface triangulations based-on 3D arbitrary point-sets are widely applied in CAGD/CAD and reverse-engineering, etc. In the first place, this paper reviews two main methods in surface triangulations, named as plane-projection and direct triangulation. For the former, Delaunay triangulations are mainly enunciated. For the later, algorithm developed by B. K. Choi is particularized. Some typical algorithms are introduced in detail, as well as various data-structures built in these algorithms. Next, since the final result of triangulation is determined by the optimal criterion, some proverbial optimal criteria are specified and analyzed in this paper, and they are thoroughly compared with each other here through anatomizing an example. It is pointed that, in practical engineering, it is necessary to develop new algorithms with new criteria for triangulations of scattered points sampled from complicated surfaces so as to maintain the properties such as better smoothness and shape preserving. Finally the time and space complexities of various algorithms are briefly and concisely discussed, also the research trend of surface triangulations based-on 3D arbitrary point-sets.

Keywords 3D arbitrary point-sets, Surface triangulations, Data structures, Optimal criteria

0 引 言

在CAGD/CAD、自由曲面设计、计算机视觉、地学等领域的许多应用中都会遇到曲面的多边形剖分, 其中最常用的也是最基本的为三角剖分. 从剖分对象来看, 剖分可分为以下3类: 平面区域剖分(二维)、三维实体剖分(三维)和空间曲面剖分, 曲面剖分问题有时也称为二维半问题.

近二、三十年来, 剖分问题由最初的平面任意点集三角剖分, 逐渐发展到对非凸边界和单、多连通域等情形下的三角剖分.

散乱点集的曲面剖分就是根据三维散乱点集构造曲面的三角剖分, 即对所给点集 $P = \{P_1, P_2, \dots, P_n\}$, 用互不相交的三角形来近似表示点集形成的曲面, 三角形的集合即为该曲面的三角剖分 T .

曲面剖分通常有两种方法, 其一是将点投影到某一平面, 运用平面的三角剖分算法完成剖分, 而三

基金项目: 东南大学优秀青年教师教学科研资助计划

收稿日期: 2002-10-29; 改回日期: 2003-06-30

维点间连接关系不变,这种方法将三维问题转化为平面问题,可称为平面投影法;其二是直接由三维点来构造剖分,称为直接剖分法。三角剖分通常有两种类型,第一类是三角剖分 T 的顶点就是所给点集 P ,不改变原点集的拓扑结构,实质是对 P 的线性插值;另外一类是在一定的误差范围内用剖分 T 来逼近曲面,这时 T 的顶点在数量和位置上均不同于原点集。本文仅讨论第一类的剖分,关于逼近剖分可参阅文献[1]~[13]。

对有 n 个点的点集,剖分中三角形个数 N 和边的总数 E 为

$$\text{开曲面: } N = 2n - n_b + 2H - 2$$

$$E = 3n - n_b + 3H - 3$$

$$\text{闭合曲面: } N = 2n - n_b + 2H - 4$$

$$E = 3n - n_b + 3H - 6$$

其中, n_b 为边界点数, H 为中间洞数。

1 平面投影法

平面投影法就是将三维点投影到某平面,如 $X-Y$ 平面,然后对投影点集作平面域的三角剖分,最终形成的曲面三角剖分的点间连接关系与相应的投影点间的连接关系相同,这种方法的实质是平面域的三角剖分。文献[14]对部分典型算法作了较为详细的综述。近几年来在剖分研究方向、具体算法等方面有了很大的进展。

当平面 4 点形成凸四边形时将有两种剖分,对采用哪一种三角剖分可以提出不同的选择原则,最常用的有最小内角最大化、边的总长度最小化即最小权等。

对最小权三角剖分有贪心算法^[15]、 $Z_{4,3}$ 算法^[16]等,贪心算法只能使边总长度尽可能小,且复杂度为 $O(n^2)$,而 $Z_{4,3}$ 算法——凸壳分层算法可以得到最小权剖分,是目前较好的算法。该算法由外向内分层计算点集的凸壳,对每一层凸多边形采用多边形三角剖分算法,直至全部完成,复杂度为 $O(n^4/m^4)$ (m 为凸壳的层数)。

按最小内角最大化原则形成的剖分也就是著名的 Delaunay 三角剖分 (DT, Delaunay Triangulation),DT 是许多领域实际应用中最受关注的一类,应用十分广泛,该剖分可追溯到 G. Voronoi 图^[17],是于 1907 年提出的,后来 Delaunay 在 1932 年提出了解决该剖分完整而实用的方法^[18],近几十年来该算法不断得到改进,以适

应不同的应用。

DT 也就是最近点意义下的 Voronoi (Dirichlet 或 Thiessen) 图的直线对偶图, Voronoi 图由许多胞元组成,每一个胞元包含点集中的一个点(每两点的垂直平分线将平面分成两部分,与某点对应的胞元实际是该点与点集中每个点垂直平分线在平面内所形成的交集),连接相邻胞元中两个点(直线段)便形成点集的 DT。DT 具有很多性质^[19,20],剖分中常用的性质有最小内角最大和最大空圆原则。

最小内角最大原则:对于一个凸四边形的两种剖分,DT 获得的两个三角形中的最小内角最大。

最大空圆原则:剖分中任一三角形的外接圆(三维为外接球面,高维为超球面)内不含有点集中的任何其他点。

最小内角最大原则又称为局部最优准则,这样获得的剖分称为局部最优(或局部等角)^[21]。

关于 DT 的算法实现有很多,大致分为以下两大类,一类是增量算法,该法从点集的某一点开始,逐步进行,每一步增加一点,直至点集为空,常用的有 Green-Sibson 算法^[22]、Bowyer 算法^[23]、Lawson 算法^[21]、Cline-Renka 算法^[24]等,这几种算法的描述详见文献[14]。另外,还有 L-S 算法、Watson 算法、以及 F-P 算法等;另一类为分治算法,该法基于分而治之的思想,每次将点集分为规模相当的两个子集,分别进行递归实现,最后拼合,目前较好的有 DeWall 算法。

1.1 增量算法

增量算法从初始的一个剖分开始,依次扫描各点,每次增加一个点形成新的剖分,在数据结构中删去用过的点,直至点集为空。

1.1.1 L-S 算法^[25]

Lawson 算法^[21],即从任意的一个剖分开始,对出现的凸四边形施加最小内角最大原则进行对角线交换,直至完成。Lawson 算法虽然能完全、正确地进行剖分,但计算量大,真正实现较为困难。

Lee 和 Schachter 继承了 Lawson 算法的边交换方法,具体实现作了改进,这种算法与 Nelson 算法^[26]相似,主要步骤如下:

- (1) 确定一个矩形区域,使所有的点位于该矩形内;
- (2) 将矩形分成大约 $n^{1/2}$ 个箱(更小的矩形域);
- (3) 从一个箱开始记录点序,即对点重排序;

(4) 置第一点于矩形中并连接此点与矩形的顶点,形成初始 DT;

(5) 加一新点于已有的 DT 中, 必有一个三角形包围该点, 连接新点与该三角形的顶点;

(6) 若上一步形成凸四边形(最多 4 个), 则依最小内角最大原则进行优化;

(7) 每个点都被用过则转第 8 步, 否则转第 4 步;

(8) 最后删去所有与初始矩形顶点有一个公共顶点的三角形, 余下的即为点集的三角剖分。

与 Lawson 算法不同的是由于有前 3 步的预处理, 搜索包围新点的三角形时的时间复杂度由原来的 $O(n)$ 减少为 $O(n^{1/2})$, 总体复杂度大约为 $O(n^{3/2})$ 。

1.1.2 Watson 算法^[27]

Watson 算法, 是一类典型的递归算法, 被广泛采用。该算法的基本思想如下: 设点集 $P_{n-1} = \{P_i; i = 1, 2, \dots, n-1\}$ 对应的 DT 为 T_{n-1} , 相应的网线集合为 E_{n-1} , 三角形集合为 t_{n-1} , 接着来构造点集 $P_n = \{P_i; i = 1, 2, \dots, n\}$ 的 DT 为 T_n 。由于任何三角剖分均由其网线集合唯一确定, 因此必须也只需考虑网线在递归算法中的生成情况, 依据相应的生成算法, 每步增加一点至 P_{n-1} , 形成 T_n , 直至所有的点都完成。

算法的主要步骤为

(1) 建立一个三角形 $\triangle E$, 包含点集中所有的点;

(2) 插入一个点, 与 $\triangle E$ 的 3 个顶点相连接形成初始剖分, 记录各三角形及其外接圆;

(3) 依次插入新点, 每次一点(见图 1)

① 计算现有剖分中外接圆包围该新点的三角形;

② 记录①产生的三角形, 其集合组成插入多边形;

③ 获得插入多边形的边界(图 1 中粗线);

④ 连接新点与插入多边形的各顶点;

⑤ 删去①、②中确定出的三角形及其外接圆(如图 1 中虚线);

⑥ 记录④产生的三角形及其外接圆;

(4) 删去所有与 $\triangle E$ 有一个公共顶点的三角形, 余下的即为点集的三角剖分。

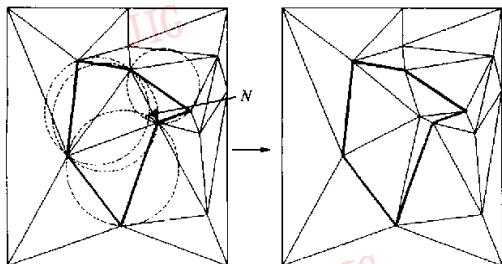


图 1 Watson 算法示意

Watson 算法的难点在于计算插入点位于一个三角形的外接圆内、外或位于外接圆上。当 4 点共圆时称为退化现象, 只有非退化时点集的 DT 才是唯一的, 故 Watson 算法中存在潜在的错误, 将影响算法的健壮性。为此, 文献[28]对确定插入点与三角形外接圆位置关系时作了 3 种不同方法的修改:

(1) 退化发生时总是将该点当作在外接圆内或外;

(2) 改变插入顺序, 即将该点放入退化点表, 待其他点完成后再插入计算;

(3) 对该插入点坐标值稍作改变再重新计算。

文献[28]认为第 2 种方法在大部分情形下相当成功, 若第 2 种方法不利时, 再用第 1 种方法, 则很少失败。

1.1.3 F-P 算法^[29]

基于一种稀疏矩阵数据结构的新算法, 即 F-P 算法从点集中心开始, 利用提出的数据结构搜索符合 DT 准则的新点, 直至全部搜索完。简要步骤如下:

(1) 排序

分别建立各点 x, y 坐标的升序列, 对应每点有一个索引对 (i, j) 表示该点 x, y 方向上的序号;

(2) 确定起始点

起始点取为 $((\max i)/2, (\max j)/2)$, 若该点不在点集中, 则以距此点最近的点作为起始点;

(3) 构造三角形

始终按最大内角原则向右寻找新点(卷包方法), 寻找时从边 AB 的右边获得一点 P , 使得 $\cos \angle APB$ 最小, 由于第 1 步实际上以各点坐标值形成了一个网格, 这样便于点的寻找。找到后即形成了第 1 个三角形, 接着按一定的方法不断确定当前应用哪一条边去寻找下一个点来构成三角形, 同时要正确处理一些特殊的情形。当一点的状态为“used”时(此时该点与余下的点不可见), 即将该点从点集中删去, 如此不断进行, 直至点集为空。

对其算法作改进, 提出了一种 UGDT 算法^[30]。UGDT 主要是将第 1 步排序改成了用一个均匀的网格来对点集进行预处理, 利用均匀网格省去点排序, 可以很方便地寻找边界边和最近点。F-P 算法能自动处理退化点和共线点, 其复杂度虽没有严格的数学证明, 但根据提供的测试例, 其复杂度为线性时间。

1.2 分治算法

分治算法是将点集分成规模相当的两部分, 对形成的两部分再进行分割, 如此递归处理, 然后对两部分作拼合计算, 直至最后全部完成。

Lewis 和 Robinson 在 1977 年提出了一种分治算法^[31],并未加证明地认为算法复杂度为 $O(n \log n)$,事实上为 $O(n^2)$,文献^[25]对该算法作了改进,改进后的算法复杂度为 $O(n \log n)$,主要子程序有点集分割和拼合。

(1)点集分割 将点集按升序排列, $P_i(x_i, y_i) < P_j(x_j, y_j)$ 当且仅当 $x_i < x_j$ 或当 $x_i = x_j$ 时 $y_i < y_j$, 将点集分为大小相当的两部分 V_L 和 V_R , $V_L = \{P_1, P_2, \dots, P_{\lfloor n/2 \rfloor}\}$, $V_R = \{P_{\lfloor n/2 \rfloor + 1}, \dots, P_n\}$;

(2)拼合 拼合前找到两个点集的上、下两公共切边(如图 2 所示),拼合是从下切边开始,从下向上进行三角形构造,直至上切边。构造三角形时按 Delaunay 准则进行。

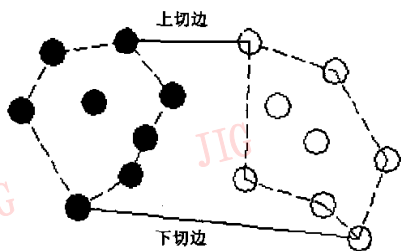


图 2 拼合示意图

另外还有多种改进的分治算法,但目前最好的为一种 E^d 空间快速算法,即 DeWall 算法^[32],该算法主要减少了拼合时的计算复杂度,在点集分割形成 V_R, V_L 的同时形成分割面 V_s ,对 V_s 用均匀网格的方法形成 DT。不仅如此,DeWall 算法还归纳了分治算法,给出了 E^d 空间分治算法的广义形式,不仅可以应用于二维,还可以对三维甚至更高维进行三角剖分,但执行复杂度没有严格的数学证明。

1.3 边界拓扑相容性

在 DT 研究中,几种实际情况引起了广泛关注,一种是对于多连通域(如中间有洞)、边界限制(如有柄等非凸情形)的拓扑结构,另一种是数量巨大的点集。

文献^[33]提出了边界限制情形的 DT 定义:

设 T 为边界限制点集 P 的一个三角剖分, $t = \triangle P_i P_j P_k \in T$, 则 T 是 DT 当且仅当不存在一个点 $P_l \in P$ 且 P_l 位于 t 的外接圆内,同时 $P_l \in W_i \cap W_j \cap W_k$, 其中, $W_i = \{P_k \in P; P_k \text{ 从 } P_i \text{ 可见(对于边界)}\}$ 。

任一点 P_i 的可见性参见图 3,点 P_j 是 P_i 的可见点,当且仅当线段 $P_i P_j$ 完全位于区域内或边界上。

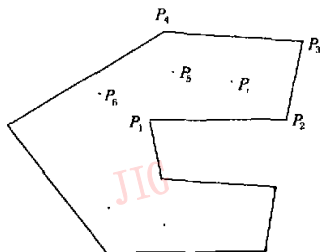


图 3 边界限制情形下点的可见性
(P_i 的可见点集 $W_i = \{P_k, k=1, 2, \dots, 7\}$)

对于边界限制点集,进行剖分时就必须考虑到边界拓扑相容问题,如图 4 所示,图中示出了一个具有非凸边界的区域及给定点和它的 Delaunay 三角形剖分,剖分不应破坏原有的拓扑结构,而图 4(b) 的三角形剖分却破坏了边界拓扑。

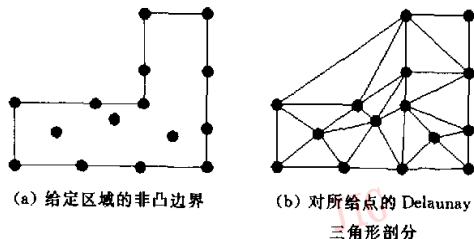


图 4 边界拓扑不相容

文献^[33]提出了一种算法,其首先对边界简单多边形进行三角剖分,然后再用增量算法进行剖分,插入新点时需计算点的可见性,其复杂度为 $O(n^2)$ 。

无论是什么形式的边界,边界的边(如图 3 中的 $P_1 P_2$)必然是点集 DT 的一条边,这个可直接由 DT 的特性推得^[29,27],由此文献^[34]提出了一种 DDT 算法,该方法属于增量算法,平均复杂度为 $O(n \log n)$ 。

2 直接剖分算法

与平面投影法不同,直接剖分算法是直接根据三维散乱点构造三角剖分。实用的直接剖分算法很少,1988 年 Choi 提出的一种增量算法^[35]是较好的一种方法,得到广泛的应用。

Choi 算法有下述几个重要概念:

(1) 顶点、边、三角形、法矢

散乱点集中的每一点在剖分中为三角形的顶点;3 个顶点 V_1, V_2, V_3 , 如果满足 $\det(V_1, V_2, V_3) \neq 0$

0 则这 3 点形成一个三角形; 从原点看去, 按逆时针方向依次连接三角形顶点, 相邻两点形成的有向边称为剖分的边; 由上述有向边按右手准则来确定三角形的单位法矢。

(2) 可见边

对于边 AB , 任一顶点 P 与它的关系有如下 3 种:

如果 $\det(A, B, P) < 0$, 则 $P. \text{Right. } AB$;

如果 $\det(A, B, P) = 0$, 则 $P. \text{On. } AB$;

如果 $\det(A, B, P) > 0$, 则 $P. \text{Left. } AB$

若顶点 $(P. \text{Right. } AB)$, 则称边 AB 相对点 P 可见, 顶点 A, B 称为 P 的可见点。

(3) 凸顶点、凸多边形

若一系列三维点形成的边封闭, 三角形不相交, 则这些点形成一个多边形。 P, Q, R 是按逆时针方向依次连接的 3 个点, 若 $\det(Q, R, P) \geq 0$, 则称 Q 是一个凸点; 若 $\det(Q, R, P) > 0$, 则称 Q 为严格凸点。 若多边形所有点都是严格凸点, 则该多边形为严格凸多边形。

(4) 轨迹生成

在工程应用中, 曲面内部某些散乱点可能属于特定的轨迹线, 如截面线、特征线等, 为了准确地反映曲面形状, 剖分中必须将这些点相互连接, 以得到相应的轨迹线。

如图 5 所示, V_1, V_2 为轨迹线上的点, 按照轨迹生成算法, V_1, V_2 必须连接, 同时去掉与其相交的边。方法是首先判断 V_1, V_2 是否已经相连, 若已相连则停止, 否则搜索与 V_1V_2 相交的边 e , 对包含 e 的四边形交换对角线即去掉了边 e 。

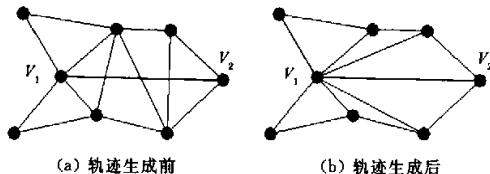


图 5 轨迹生成

(5) 边界剪裁

边界剪裁就是要从剖分中删去外边界环之外和内边界环之内“不合法”的三角形, 对图 4 中的例子,



图 8 主剖分过程示意

首先用轨迹生成算法获得满足要求的剖分, 如图 6 中(a)所示, 然后进行边界剪裁, 去掉图 6(a)中标号为 1、2、3 的 3 个三角形, 最终获得正确的剖分如图 6(b)所示。

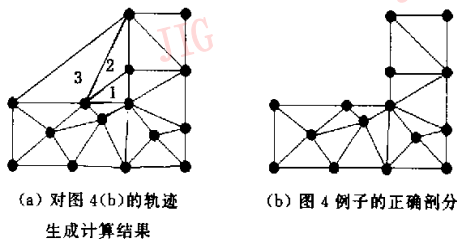


图 6 轨迹生成和边界剪裁

Choi 算法的简要步骤如下:

第 1 步 预处理

① 找到一个点 P_c , 使得从 P_c 点能够看到所有的散乱点, 并作一过 P_c 的轴 u ;

② 对所有的点做变换 $P_i \rightarrow V_i, V_i = P_i - P_c$;

③ 按 V_i 与 u 的夹角大小, 将 V_i 升序排列。

第 2 步 三角剖分

① 初始三角剖分

最初的剖分是从 V_1, V_2 开始, 首先得到边 V_1V_2 , 接着从 V_3 开始逐点搜索, 若共面则继续搜索, 直至一不与前面点共面的顶点, 连接此点与前面所有点从而构成初始剖分, 如图 7 所示(图中点 1、2、3、4 共面, 5 与它们不共面)。

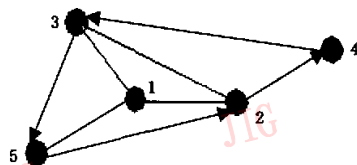
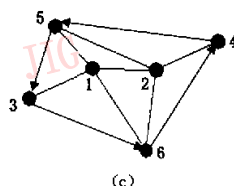


图 7 初始三角形剖分

② 剖分主过程

(a) 下一个新点 N 插入时, 搜索 N 的所有可见点(图 8(a)), 并确定最左可见点 $LVN(V_3)$ 和最右可见点 $RVN(V_4)$;



(b)构造一个三角形,它的顶点是 N 、 LVN 和 LVN 的下一点(图8(b)中 $\triangle V_6V_3V_1$);

(c)若此时边界非凸,则继续从此非凸点开始与其下一点、 N 构造三角形(图8(c)中 $\triangle V_1V_6V_2$),并从TBL中删去该非凸点,直至边界为凸(图8(c));

(d)处理完一个点,则将此点从点集中删除,若点集为空,则停止,否则转步骤(a)。

第3步 轨迹生成。

第4步 边界剪裁。

3 曲面剖分常用数据结构

剖分的具体实现依赖于一定的数据结构,对某一种剖分而言,合理的数据结构可以方便地编程,下面简要地叙述3种常用的数据结构。

(1) 分区栈结构(region stack)^[31]

这种数据结构是Lewis和Robinson在1977年提出一种分治算法时采用的,它在一个栈中存储区域连续分割信息。对每一个分区而言,都包含首尾内部数据、边界点和边界点数。分割点集时,算法访问栈中最后的一段信息,将之分割,并对分割后的点重新排序,再将分割后两部分存回栈中以取代原有的那段信息,这样可通过递归实现。

(2) 稀疏矩阵(sparse matrix)结构^[29]

这是F-P算法采用的数据结构,它由3个结构组成(C代码):

```
struct xynode
{
    double x_value; //x 值
    int numbers_of_points; //具有相同x值的点数
    LIST ptr;
}
//LIST
struct vnode
{
    int index; //y 索引值
    int used; //标志,若为“used”则该点不再参加剖分
    POINT previous, next;
}
typedef struct vnode * LIST;
// POINT
struct pnode
{
    int i, j; //点的x、y索引值
}
```

```
typedef struct pnode * POINT;
```

这种数据结构是在对数据预处理时完成的,分别对点 (x_i, y_i) 的 x_i 和 y_i 从小到大排序,建立 x 、 y 方向的索引序列,相同的坐标值具有相同的索引号,这样每一个点对应一个索引对 (i, j) ,然后将数据添入稀疏矩阵中,如图9所示,图中表示了有两个有相同横坐标的点的表示方法。

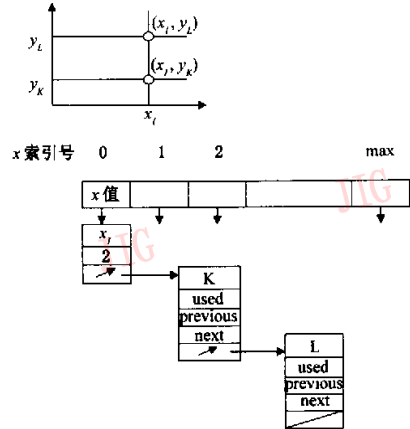


图9 稀疏矩阵数据结构

这种稀疏矩阵数据结构为F-P算法的最近点确定和执行向右卷包计算带来了极大的方便,当F-P算法改进成UGDT算法时,稍微地对它进行了改变,但其实质是一样的。这种结构具有较小的时空代价,为稳定可靠地运行剖分算法奠定了基础。

(3) LTL (Lawson's Triangle List) 和 TBL (Transient Boundary List)

LTL是由Lawson在进行曲面 C^1 插值时提出的三角形表数据结构^[36],TBL是由Choi等人提出的^[35]。在直接剖分算法中往往将Lawson三角形表(LTL)和临时边界表TBL结合使用,LTL记录三角形生成信息,在新点插入时,动态改变TBL可以即时更新LTL,而LTL和TBL的结合使用可以方便地进行边界剪裁和轨迹生成计算。

LTL和TBL的结构可参见图10,其中(a)、(b)、(c)反映的是图8中(a)、(b)、(c)中三角形的更新情况。LTL中有两组数据,第1组记录的是三角形的3个顶点,第2组的3个数据分别记录对应顶点的对边右边相邻的三角形(按逆时针方向),无相邻三角形时则为0,边界剪裁时去除剖分中边界边右边的三角形(无论是内边界还是外边界);TBL是由当前剖分的外边界点组成,每一个节点有两个数

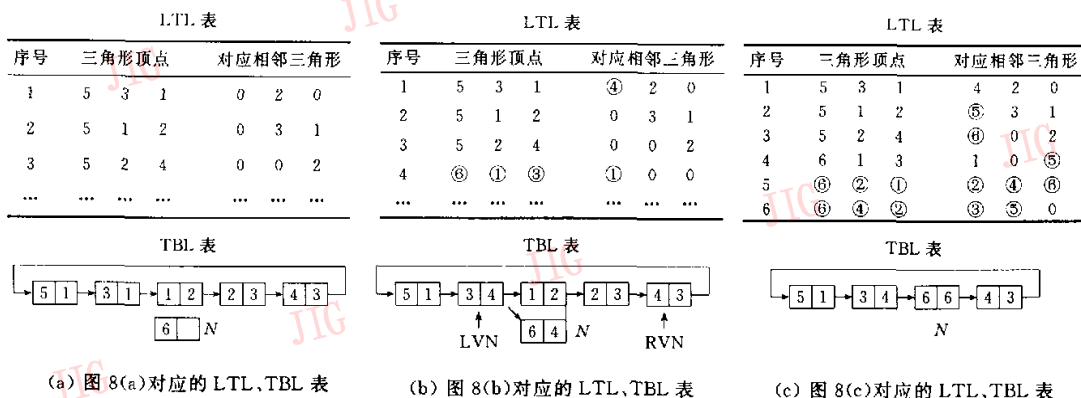


图 10 图 8 对应的 LTL 和 TBL 表

据,分别表示边界点序号和该边界点与其下一个点形成的有向边右边的三角形序号,每插入一个新点时,重新形成新的 TBL, TBL 的每一步更新会形成一个新的三角形,这时需立即更新 LTL.

4 曲面剖分优化准则

剖分必须依据一定的优化准则,优化准则的选取直接影响到剖分结果.当剖分的两个三角形形成一个凸四边形时,存在两种剖分形式,这时须依据一定的准则来确定.曲面剖分常用的优化准则有外接曲面拟圆准则、标量积最大、空间形状优化准则以及光顺准则等,其中外接曲面拟圆准则是单侧性准则和模拟圆准则的广义形式,它和标量积最大准则只单纯考虑形成凸四边形的两个三角形之间的关系,而空间形状优化准则以及光顺准则则考虑两个三角形周围的 4 个三角形对剖分的影响.

4.1 外接曲面拟圆准则

外接曲面拟圆准则^[37]是二维剖分中的外接圆准则(Delaunay)和三维的外接球准则在曲面剖分中的推广形式,也是单侧性准则和模拟圆准则的广义形式.

单侧性准则称:曲面剖分的三角形的一侧不含有点集中的任一点,则称该三角形为局部最优;

模拟圆准则称:若三角形的一侧没有满足如下条件的点:在点集中且向三角形平面的投影落在该三角形的外接圆内,则称该三角形为局部最优.

由于这两个准则与外接圆准则(Delaunay)和三维的外接球准则不能统一描述,且在避免病态三角形方面尚有不足,因此文献[37]提出了外接曲面拟圆准则.

外接曲面拟圆:给定曲面 Σ 上不共线的 3 点 P_1 、 P_2 、 P_3 ,它们确定的平面的单位法矢为 \mathbf{n} ,以三角形 $\triangle P_1P_2P_3$ 的外接圆圆心为起点沿 \mathbf{n} 方向可以确定一射线,该射线与 Σ 的交点为 C_0 (若有多个交点则取最远处点),以 C_0 为中心,以 $\|C_0P_1\|$ 为半径的球与 Σ 的交称为 $\triangle P_1P_2P_3$ 关于 Σ 的外接曲面拟圆.

外接曲面拟圆准则:曲面上 3 点组成三角形,若外接曲面拟圆区域内没有内环且不含有集中任意点,则称该三角形局部最优.

文献[37]同时还证明了以下两个结论:

- (1) 满足外接曲面拟圆准则必满足模拟圆准则;
- (2) 外接曲面拟圆准则等价于单侧性准则.

4.2 标量积最大准则

标量积最大准则^[38]是指对剖分形成的凸四边形,存在的两种剖分,计算每一剖分两个三角形单位法矢的标量积,取最大的一个剖分.标量积最大也就是形成剖分的两个三角形平面夹角最大,更趋于平坦.

4.3 空间形状优化准则

空间形状优化准则^[39]是指剖分形成的两个三角形有 5 条棱边,根据周围 4 个相邻的三角形判断各棱边的凹凸性,凸者边值为 1,凹者边值为 -1,共面时边值为 0,计算两种情形下的 5 条边边值之和的绝对值 K ,取 K 值大的一种为最优剖分.具体如图 11 所示,图中 l_{ij} 表示三角形 i 与三角形 j 的公共棱边的凹凸,公共棱边为凸时,其边值为 1;公共棱边为凹时,其值为 -1;两个三角形共面时,其值为 0.接着分别计算 K_a (图 11(a))和 K_b (图 11(b)):

$$K_a = |l_{12} + l_{25} + l_{26} + l_{14} + l_{13}|$$

$$K_b = |l_{12} + l_{25} + l_{26} + l_{14} + l_{13}|$$

比较它们的大小,取 K 值较大的那种剖分.

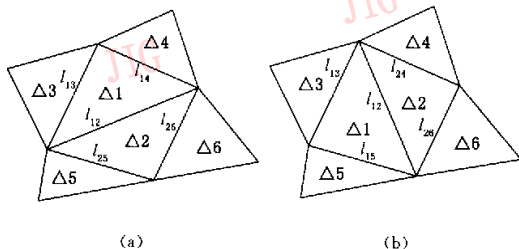


图 11 空间形状优化准则

线性插值的保凸性和光顺性会直接影响插值曲面保凸和光顺的程度,空间形状优化准则的出发点就是由剖分生成各类连续曲面时,生成的曲面比较光顺。

4.4 光顺准则

光顺准则^[45]是 Choi 算法中采用的剖分准则,剖分中两个三角形形成的凸四边形的每条边为两个三角形所共有,分别计算这两个三角形单位法矢标量积,这样对应每一种剖分就有 4 个标量积值,计算 4 个标量积的最小值,取最小值最大的一种剖分作为最优剖分。

5 讨论

5.1 算法

计算平面散乱点集 DT 的时间复杂度为 $O(n \log n)$,对于常用的两种算法:分治算法和增量算法,它们各有优缺点,增量算法复杂度通常为 $O(n^2)$,但编程容易,所需存储空间较少,分治算法则需要更多的存储空间,但计算速度较快,复杂度为 $O(n \log n)$,前者在处理海量数据点集时,执行时间成为障碍,后者对计算硬件要求较高。

散乱点集的曲面剖分必须进行轨迹生成和边界剪裁等计算,而后再考虑编程简单等因素,常采用增量算法,如 Watson 算法易于递归实现;F-P 算法由于采用了稀疏矩阵数据结构,使得编程简单;UGDT 算法由于使用了均匀网格,大大简化了几何搜索;而直接剖分 Choi 方法在寻找能够看到所有散乱点的点 P_i 时,有时显得十分困难,尽管如此,由于其合理地应用了 LTL 和 TBL,能方便地进行轨迹生成、边界剪裁等计算,同时也便于施加各类优化准则,因此是一种很好的曲面剖分算法。

在实际应用中,往往是将几种方法结合,如文献[40]结合了文献[24]的数据预处理和轨迹生成与边界剪裁等拓扑重整算法;文献[41]将 F-P 算法与拓

扑重整算法相结合,提出了对单连通域的剖分算法;文献[38]在处理多值曲面的剖分时,将问题分解为非多值曲面的剖分,然后拼合,而非多值曲面的剖分则完全按 Choi 算法。

目前,曲面剖分研究主要是针对散乱点集的交互式、全自动的网格剖分^[4,42~44],以及多维数据点集的三角剖分以形成三维深度图象^[45],但核心算法不外乎基本的增量算法和分治算法。

5.2 优化准则

剖分优化准则直接影响到剖分结构,从计算机辅助几何设计、计算机图形学的曲面设计和连续插值曲面视察的角度来看,考虑几何上的保形性和光顺要求时,基于 Delaunay 三角剖分准则在离散点曲面剖分中已经不再适用,那么,什么样的优化准则能满足并具有更好的一般性;能否直接从几何光顺定义来确定优化准则;前述的几种准则是否适用,下面将举例来说明。

如图 12 所示,图中取曲面 $Z = -x^2 + 5$ 的 8 个点作为例子,该例形成两种剖分图 12(a)和图 12(b),对该例而言,从保形和光顺的角度看,应采用图 8(b)形式剖分,具体分析见表 1。由表 1 看出,根据空间形状优化准则,计算得 $K_a = K_b = 1$,两种 K 值相等,根据该准则两种剖分皆可;根据两剖分三角形标量积最大准则,则应采用图 12(a)剖分;根据光顺准则,两种情形下的单位法矢标量积最小值相等,两种剖分皆可。标量积最大准则试图避免过于尖锐

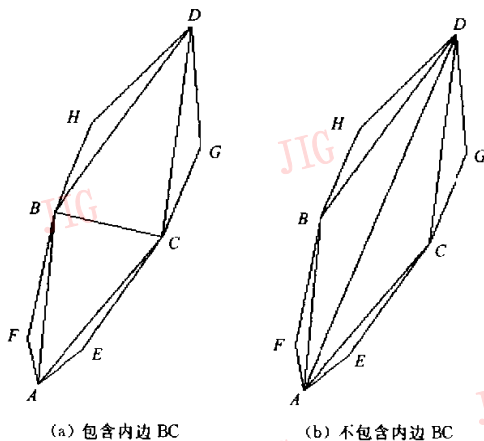


图 12 三角剖分优化准则举例

(8 点 $A \sim H$ 取自曲面 $Z = -x^2 + 5$, $A(0, -5, 5)$, $B(-1, 0, 4)$, $C(1, 0, 4)$, $D(0, 5, 5)$, $E(0.5, -3.75, 4.75)$, $F(-0.5, -3.75, 4.75)$, $G(1, 2.5, 4)$, $H(-1, 2.5, 4)$)

表 1 优化准则举例分析

棱边	剖分($\triangle ACB/\triangle BCD$)				棱边	剖分($\triangle ACD/\triangle ADB$)			
	三角形对	法矢	单位法矢标量积	l 值		三角形对	法矢	单位法矢标量积	l 值
AB	$\{\triangle ACB, \triangle ABF\}$	$\{(0,1,5), (0,1,5)\}$	1	0	AB	$\{\triangle ADB, \triangle ABF\}$	$\{(-1,0,1), (0,1,5)\}$	$\frac{5}{2\sqrt{13}}$	-1
AC	$\{\triangle ACB, \triangle AEC\}$	$\{(0,1,5), (0,1,5)\}$	1	0	BD	$\{\triangle ADB, \triangle BDH\}$	$\{(-1,0,1), (-1,0,1)\}$	1	0
CD	$\{\triangle BCD, \triangle CGD\}$	$\{(0,-1,5), (1,0,1)\}$	$\frac{5}{2\sqrt{13}}$	1	AC	$\{\triangle ACD, \triangle AEC\}$	$\{(1,0,1), (0,1,5)\}$	$\frac{5}{2\sqrt{13}}$	-1
BD	$\{\triangle BCD, \triangle BDH\}$	$\{(0,-1,5), (-1,0,1)\}$	$\frac{5}{2\sqrt{13}}$	1	CD	$\{\triangle ACD, \triangle CGD\}$	$\{(1,0,1), (1,0,1)\}$	1	0
BC	$\{\triangle ACB, \triangle BCD\}$	$\{(0,1,5), (0,-1,5)\}$	$\frac{12}{13}$	-1	AD	$\{\triangle ACD, \triangle ADB\}$	$\{(1,0,1), (-1,0,1)\}$	0	1

的两个三角形出现,计算简单,但在保形性和光顺性方面不尽人意;空间形状优化准则主要着眼于保形性;光顺准则主要着眼于曲面的光顺性。

使用空间形状优化准则和光顺准则都要考虑周围三角形的影响,因此增加了计算量,更为重要的是对开曲面,由于边界的存在,尤其是各种情形限制的非凸边界,致使应用存在困难。应用这两个准则时,需判断边的凹凸性和计算边左右两个三角形法矢的标量积,当边处于边界时,该边只属于一个三角形,则上述计算无法进行,这时必须扩充边界即构造“伪边界”,使该边成为内边,这样才能完成计算。

在实际应用中,如逆向工程中的曲面重构,空间散乱点是对实际曲面采样得到的,而实际曲面有时十分复杂,剖分不但要求光顺特性,还需正确反映曲面形状,如在两个曲面相交处,如何使得剖分自动保持这些形状,这就是保形问题。光顺和保形与曲面的内在特性密切相关,如切平面、曲率等,而曲面剖分的优化准则则直接决定了剖分的最终结构,但现有的优化准则无法满足要求。从曲面的内在特性着眼,要获得光顺和保形特性的三角剖分需寻找新的剖分优化准则。

综上所述,根据三维散乱点进行曲面剖分涉及到许多方面,目前的主要研究方向有:

(1) 在算法的选择方面,主要考虑执行时间复杂度、编程效率,还有与算法匹配的、简洁合理的数据结构,以减少几何搜索的时间,当处理海量数据点集时,还要考虑到硬件的配置。

(2) 对各种边界限制情形,多维数据点集的交互式、全自动的网格剖分,在交互环境下能自动处理数据点的增、删、修改后的三角剖分。

(3) 由于优化准则决定了剖分结果,而在 CAD/CAGD 等许多应用中,由剖分构造曲面时,往往着眼于曲面的光顺和保形特性,从这个角度确立新的剖分优化准则还有待进一步探索。

参考文献

- Hoppe H, Deroose T, Duchamp T *et al.* Surface reconstruction from unorganized points [A]. In: Proceedings of SIGGRAPH [C], Chicago, Illinois, 1992; 71~78.
- Curless B, Levoy M. A volumetric method for Building complex models from range images [A]. In: Proceedings of SIGGRAPH [C], New Orleans, Louisiana, 1996; 303~312.
- 吕秋灵,沈丽宁. 一种改进的数据网格化方法[J]. 计算机辅助工程, 1996,5(1):56~58.
- Xu X, Pain C C, Gaddard A J H *et al.* An automatic meshing technique for Delaunay triangulations [J]. Computer Methods Applied Mechanics & Engineering, 1998,161(3):297~303.
- Vigo M, Pla N, Brunet P. Directional adaptive surface triangulation [J]. CAGD, 1999,16(1):107~126.
- Weimer H, Warren J. Fast approximating triangulation of large scattered datasets [J]. Advances in Engineering Software, 1999, 30(4):389~400.
- Vigo M, Pla N. Computing directional constrained Delaunay triangulations [J]. Computer & Graphics, 2000,24(2):181~190.
- Balan R, Taubin G. 3D mesh geometry filtering algorithms for progressive triangulation schemes [J]. CAD, 2000,32(13):825~846.
- Bertram M, Barnes J C, Hamann Bernd *et al.* Piecewise optimal triangulation for the application of scattered data in the plane [J]. CAGD, 2000,17(7):767~787.
- Sheffer A. Model simplification for meshing using face clustering [J]. CAD, 2001,33(13):925~934.
- Dey T K, Giesen J, Hudson J. Delaunay based shape reconstruction from large data [A]. In: IEEE Proceedings Symposium On Parallel and Large-Data Visualization and Graphics [C], San Diego, California, 2001:19~27.
- 邱泽阳,肖双九,杨海成等. 一种基于逼近理论的曲面重构方法 [J]. 计算机辅助设计与图形学学报, 2001,13(7):622~625.

- 13 刘伟军, 孙玉文, 刘健. 点云数据的曲面重构新算法[J]. 工程图学学报, 2001, 22(2): 47~52.
- 14 周晓云, 朱心雄. 散乱数据点三角剖分方法综述[J]. 工程图学学报, 1993, 13(1): 48~54.
- 15 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: 101~103.
- 16 周培德. 平面点集三角剖分的算法[J]. 计算机辅助设计与图形学学报, 1996, 8(4): 259~264.
- 17 Voronoi G. Nouvelles applications des paramêtres continues à la théorie des formes quadratiques. Premier Mémoire: Sur quelques propriétés des formes quadratiques positives parfaites [J]. J. reine angewandte Mathematik, 1907, 133(1): 97~178.
- 18 Delaunay B. Neue Darstellung der geometrischen krystallographie [J]. Zeitschrift krystallographie, 1932, 84(1): 109~149.
- 19 Sibson R. Locally equiangular triangulations [J]. The Computer Journal, 1978, 21(3): 243~245.
- 20 Lawson C L. Properties of n-dimensional triangulations [J]. CAGD, 1986, 3(2): 231~246.
- 21 Lawson C L. Generation of a triangulation grid with application to contour plotting [M]. Technology Memorandum # 299, California Institute of Technology Jet Propulsion Laboratory, 1972.
- 22 Green P J, Sibson R. Computing Dirichlet tessellations in the plane [J]. The Computer Journal, 1978, 21(2): 168~173.
- 23 Bowyer A. Computing Dirichlet tessellations [J]. The Computer Journal, 1981, 24(2): 162~166.
- 24 Cline A K, Renka R L. A storage-efficient method for construction of a Thiessen triangulation [J]. Rocky Mount Journal of Mathematics, 1984, 14(1): 119~139.
- 25 Lee D T, Schachter B J. Two algorithms for construction a Delaunay Triangulation [J]. Int'l. Journal Computer & Information Sciences, 1980, 9(3): 219~243.
- 26 Nelson J M. A triangulation algorithm for arbitrary planar domains [J]. Applied Mathematical Modeling, 1978, 2(3): 151~159.
- 27 Watson D F. Computing the n-dimensional Delaunay tessellation with applications to Voronoi polytopes [J]. The Computer Journal, 1981, 24(2): 167~172.
- 28 Field D A. Implementing Watson's algorithm in three dimensions [A]. In: Proceedings of the 2nd annual ACM Symposium Computational Geometry [C], New York, 1986: 246~259.
- 29 T-P Fang, Piegl L A. Algorithm for Delaunay triangulation and convex-hull computation using a sparse matrix [J]. CAD, 1992, 24(8): 425~436.
- 30 Fang T P, Piegl L A. Delaunay triangulation using a uniform grid [J]. IEEE Computer Graphics and Applications, 1993, 13(5): 36~47.
- 31 Lewis B A, Robinson J S. Triangulation of planar regions with applications [J]. The Computer Journal, 1978, 21(2): 324~332.
- 32 Cignoni P, Montani C, Scopigno R. DeWall: A fast divide and conquer Delaunay triangulation algorithm in Ed [J]. CAD, 1998, 30(5): 333~341.
- 33 Floriani L D, Falcidieno B, Pienovi C. Delaunay-based representation of surfaces defined over arbitrarily shaped domains [J]. Computer Vision, Graphics, and Image Processing, 1985, 32(1): 127~140.
- 34 Sapidis N, Perucchio R. Delaunay triangulation of arbitrarily shaped planar domains [J]. CAGD, 1991, 8(2): 421~437.
- 35 Choi B K, Shin H Y, Yoon Y I *et al.* Triangulation of scattered data in 3D space [J]. CAD, 1988, 20(5): 239~248.
- 36 Lawson C L. C^1 surface interpolation for scattered data on a sphere [J]. Rocky Mount. J. Math., 1984, 14(1): 223~237.
- 37 孔德慧, 陈其明, 汪叔淳. 一种新的曲面剖分优化准则 [J]. 工程图学学报, 1995, 16(1): 28~34.
- 38 柯映林, 周儒荣. 实现 3D 离散点优化三角剖分的三维算法 [J]. 计算机辅助设计与图形学学报, 1994, 6(4): 241~248.
- 39 姜寿山, 杨海成, 候增选. 用空间形状优化准则完成散乱数据的三角剖分 [J]. 计算机辅助设计与图形学学报, 1995, 7(4): 241~249.
- 40 周晓云, 何大曾, 朱心雄. 实现平面上散乱数据点三角剖分的算法 [J]. 计算机辅助设计与图形学学报, 1994, 6(4): 256~259.
- 41 管储, 孟宪琦, 魏新民. Delaunay 三角网格化算法及实现 [J]. 西北工业大学学报, 1996, 14(1): 138~142.
- 42 Schroeder W J, Shepard M S. Geometry-based fully automatic mesh generation and the Delaunay triangulation [J]. Int'l Journal Number Mathematics Engineering, 1988, 26(11): 2503~2515.
- 43 Zhu J Z, Zienkiewicz O C, Hinton E *et al.* A new approach to the development of automatic quadrilateral mesh generation [J]. Int'l Journal Number Mathematics Engineering, 1991, 32(4): 849~866.
- 44 Floriani L D, Magillo P, Morando F *et al.* Dynamic view-dependent multiresolution on a client-server architecture [J]. CAD, 2000, 32(13): 805~823.
- 45 Weimer H, Warren J, Trouter J *et al.* Efficient co-triangulation of large data sets [A]. In: IEEE Proceedings of Visualization '98 [C], North Carolina, 1998: 119~126.



张永强 1968年生, 2002年于南京工业大学获材料科学硕士学位, 现为东南大学博士研究生. 主要研究方向为 CAGD/CAD、计算机仿真等.



达飞鹏 1968年生, 1996年于东南大学获控制理论与控制工程博士学位, 现为东南大学教授. 主要研究方向为逆向工程、三维模型重构、智能控制等.



宋文忠 1936年生, 东南大学教授, 博士生导师, 1960年毕业于东南大学动力工程系. 主要研究方向为先进控制理论、三维模型重构等.