

# 一种新的带特征约束的 Delaunay 三角剖分算法

梅承力,肖高逾,周源华

(上海交通大学图像通信与信息处理研究所,上海 200030)

**摘要:** 本文提出了一种新的带特征的约束条件的散乱数据 Delaunay 三角剖分算法. 本算法通过在特征线上加入若干数据点,使得在保留特征线的同时,最终的剖分结果也满足全局 Delaunay 特性. 与现有同类算法相比,本算法更充分地利用了 Delaunay 三角剖分中的圆准则,从而大大减少了所需加入的附加点数. 实验结果证实了本算法的这一特性.

**关键词:** Delaunay 三角剖分; 特征约束; 全局 Delaunay 特性; 圆准则

**中图分类号:** TP391.41 **文献标识码:** A **文章编号:** 0372-2112 (2001) 07-0895-04

## A New Algorithm for Conforming Delaunay Triangulation

MEI Cheng-li, XIAO Gao-yu, ZHOU Yuan-hua

(Institute of Image Communication & Information Processing, Shanghai Jiaotong Univ., Shanghai 200030, China)

**Abstract:** A new algorithm for Conforming Delaunay triangulation in 2-D is presented. The algorithm inserts points along constraint edges in order to maintain the global Delaunay criterion, while at the same time preserves the constraint edges. Compared with other strategies, our algorithm makes better use of the empty circle property in Delaunay triangulation, so that the number of extra points is greatly reduced. Corroborating experimental results are also given.

**Key words:** delaunay triangulation; constraint; global delaunay criterion; criterion

### 1 引言

在计算机图形学和计算机辅助设计中,经常需要通过一系列散乱数据点来重建物体的表面形态. 在各种表面重建策略中,基于三角剖分的三角网格表示法由于其在刚性变换下的不变性、对数据分布密度变化的良好适应性、以及便于局部更新而成为比较常用的一种方法. 对散乱数据进行三角剖分可按不同的规则进行. 其中 Delaunay 三角剖分由于它的结果尽可能地避免了过于狭长和尖锐的三角形而受到广泛的研究<sup>[1,2]</sup>. 许多情况下,在给定了—系列数据点后,还可能要规定若干条连接部分点的边,并要求这些边必须在剖分结果中出现,这一类问题被称为带特征约束的 Delaunay 三角剖分. De Floriani<sup>[3]</sup>对此进行了研究并提出了解决方案,Anglada<sup>[4]</sup>在此基础上进行了一定的改进,但是其最终的剖分结果并不能满足全局 Delaunay 特性. Faugeraus<sup>[5]</sup>提出了在特征线上增加数据点以满足全局 Delaunay 特性的思想,但其算法所加入附加点的数目较多,且当特征线周围存在数据点时无能为力. Fang<sup>[6]</sup>的方法虽然没有这一问题,但它要求所有特征线必须事先一次给定,不允许逐条加入,并且剖分结果会因初始点选取的不同而变化. 此外,文献[7]提出了对特征线进行对半划分而加入附加点的思想,但是取特征线中点这一操作本身即存在着一定的盲目性,因而需加入的附加点的数目仍较多. 本

文提出了一种新的带特征约束的 Delaunay 三角剖分算法,该方法不仅允许特征线动态加入,而且更充分地利用了圆准则特性,从而大大减少了所必需的附加点数.

### 2 新型带特征约束的 Delaunay 三角剖分算法原理

Delaunay 三角剖分结果的最大优点是它尽可能避免了过于尖锐和狭长的三角形的出现,即最终所有的三角形都满足最大最小角准则 (Max-Min Angle Criterion): 在三角网格中,对任意相邻并构成严格凸的四边形的两个三角形来说,其六个内角中的最小值大于交换四边形的对角线而形成的另两个三角形的六个内角中的最小值. 这一准则的一个等价表述即是圆准则 (Circle Criterion): 在三角形网格中,任意三角形的外接圆将不包含除该三角形以外的其它任何三角形的顶点. 本算法即是在充分利用圆准则特性的基础上而提出的.

下面结合图例来说明本算法的基本原理. 图 1 为特征线只穿过两个三角形的情况. 图中  $A, B, C, D$  为已知的四个数据点,其在无约束下的 Delaunay 三角剖分结果为  $ABD$  和  $BCD$ . 现要求在  $A$  和  $C$  之间加一条连接  $A, C$  的直线作为特征线. 为此需要在  $A, C$  之间另外加入一点  $P$ ,并希望在新的 Delaunay 剖分结果中包含线段  $AP$  和  $PC$ ,即  $ABP, ADP, BCP$  和  $CDP$  均满足 Delaunay 圆准则. 若  $P$  位于  $ABD$  的

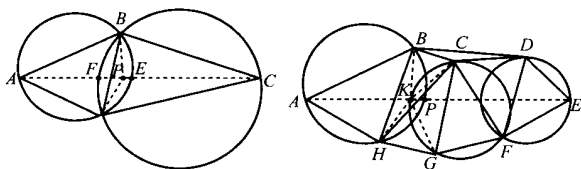


图 1 特征线穿过两个三角形时的处理

图 2 特征线穿过多个三角形时的处理

外接圆内且在  $BCD$  以内, 则对于  $BCD$  来说, 相当于在其内增加了一点  $P$ . 依照 De Florian<sup>[8]</sup> 的思想, 该三角形先被分割成  $BCP$ ,  $CDP$  和  $BDP$  三部分. 此时, 显然  $ABD$  和  $BDP$  的剖分方式不能满足圆准则, 而对四边形  $ABPD$  做对角线交换 (edge swap) 的所得到的结果  $ABP$  和  $ADP$  则满足这一要求. 这样由于  $P$  点的加入, 新的 Delaunay 剖分的结果就成为  $ABP$ ,  $ADP$ ,  $BCP$  和  $CDP$ , 从而  $A, C$  之间的连线得以保留下来. 同理, 当  $P$  位于  $BCD$  的外接圆内且在  $ABD$  之内时也能够得到同样的结论. 综上, 只要新加入的点  $P$  在两个三角形的外接圆与特征线的交点  $E$  和  $F$  之间, 则该特征线就能够在最终的剖分结果中保留. 在实际操作中, 为了得到三角形形状尽可能规则的剖分结果, 取  $P$  点为  $E$  和  $F$  的中点即可. 需要注意的是,  $P$  点的引入可能会影响到分别与  $AB, BC, CD$  和  $DA$  四条边相邻的三角形的 Delaunay 特性. 所以还需要分别对同这四条边邻接的三角形进行进一步检验.

图 2 为特征线穿过多于两个三角形的情况. 图中数据点  $A, \dots, H$  在无约束下的剖分结果如实线所示. 现要求新增特征线  $AE$ . 为此先由  $A$  至  $E$  顺序找出并记录下所有与  $AE$  相交的三角形序列  $(1, \dots, m)$ . 借用 De Florian 的概念, 将由这些三角形所组成的区域称为该特征线的影响域 (influence region). 随后计算第一个三角形 ( $ABH$ ) 的外接圆与  $AE$  的交点  $P$ , 看其是否落在最后一个三角形 ( $DEF$ ) 的外接圆内. 若  $P$  在此圆内, 则只需将这两个外接圆与特征线交点的中点作为  $P$  点新的坐标, 连接  $P$  与所有与  $AE$  相交的三角形的顶点, 记录下这些新产生的三角形并检测它们是否会影响到原有剖分结果的 Delaunay 特性即可. 否则, 先找出包含  $P$  的三角形 (设其在三角形序列中的序号为  $n$ ), 再作其后一个 (第  $n+1$  个) 三角形的外接圆, 若其与  $AE$  的交点  $K$  在线段  $AP$  之内, 则将  $P$  点移至  $K$  处. 随后, 连接  $P$  与三角形序列中前  $n$  个三角形的顶点, 将  $ABP, BCP, GHP$  和  $AHP$  记作新生成的三角形. 由于此时  $P$  点位于  $ABH$  的外接圆内, 所以  $ABP$  和  $AHP$  显然满足 Delaunay 条件. 又由于  $ABH, BCH$  和  $CHG$  本身即是无约束条件下的 Delaunay 三角剖分的结果, 所以它们相互之间也都满足 Delaunay 约束条件. 需要注意的是, 这时并不知道  $P$  点的加入是否会影响那些与  $AB, BC, AH$  和  $HG$  边相邻的三角形的 Delaunay 特性. 所以还要对这些三角形的 Delaunay 特性进行检验. 具体的处理过程将在下一节中描述. 此时由  $CPG$  和第  $n+1$  至第  $m$  个三角形所组成的三角形序列显然满足无约束的 Delaunay 特性, 因此可以对由  $P, C, \dots, G$  所组成的点集重复执行以上步骤. 需要注意的是此时特征线的两端点已变为  $P, E$ . 将这一操作递归地

进行下去, 直至该条特征线处理完毕. 最后再检验所有新产生的三角形是否影响到与之相邻的原有三角形的 Delaunay 特性.

由以上分析可以看出, 本算法在特征线上添加附加点时, 是从特征线的一个顶点开始, 利用该顶点所在的三角形的外接圆与特征线的交点而确定附加点的位置. 随后, 这一新加入的点又被视为新的顶点, 并重复以上的操作, 直至整条特征线被处理完毕. 整个过程中, 在每次确定附加点的位置时都尽可能地利用了圆准则特性, 因而减少了附加点的个数. Fang 的算法虽然在加入附加点时也利用了圆准则特性, 但却并没有将这一思想贯彻于整个算法的始终. 并且 Fang 的方法的判断和处理过程也较为繁杂. 从稍后的对比实验中将可以看到, 若采用 Fang 的算法, 所需要的附加点的个数要比使用本算法时多. 而且, 由于本算法对所有特征线的处理是逐条进行的, 所以这些约束条件无需事先全部一次给定, 而是可以在处理过程中逐渐加入. 这也是本算法的另一个优越之处.

### 3 算法的实现

如上节所述, 对一条特征线的影响域重新做三角剖分的结果可能会改变与影响域相邻的三角形的 Delaunay 特性. 因此为了完成最终的 Delaunay 三角剖分, 还有必要在剖分过程中采取特殊的处理步骤. 首先, 建立待处理特征线的集合, 并将特征线存入一个集合中. 不妨将这个集合记为 `constraint`. 同时, 由于每条特征线在处理完后都被分割成若干条新的短特征线段, 所以还要另外建立一个用于存放这些新的特征线的集合. 再将这个集合记为 `constraint_finish`.

当每条特征线处理完毕后, 检测所有新产生的三角形是否破坏了与之相邻的三角形的 Delaunay 特性. 若是, 则一方面递归地对其它三角形进行相同的检测, 另一方面看这些因为对角线交换而改变了的边是否属于已被处理过的特征线. 如果它们是已处理过的特征线, 则将这些边从已处理特征线的集合中取出, 并重新放入待处理特征线的集合, 这一过程可用伪码表示为:

```
Procedure Edge_swap_triangle (T, constraint, constraint_finish)
```

```
// T: 对影响域的三角剖分结果.
```

```
Stack := EmptyStack;
```

```
将 T 中的全部三角形压入 Stack 中;
```

```
While Not Empty(Stack) do
```

```
  t := Pop(Stack);
```

```
  令三角形 t 的三个顶点为  $P_1, P_2, P_3$  (其中  $P_3$  为新加入的特征点),
```

```
  t.opo 为与  $P_3$  相对的三角形. 设 t.opo 的三个顶点为  $P_1, P_2, P_4$ .
```

```
  If  $P_4$  在 t 的外接圆内 then
```

```
    对 t 和 t.opo 这两个三角形做边交换 (edge swap);
```

```
    Push(Stack,  $P_1P_4P_3$ );
```

```
    Push(Stack,  $P_2P_4P_3$ );
```

```
  If  $P_1P_2$  constraint_finish then
```

```
    将  $P_1P_2$  从 Constraint_finish 中取出, 并放入
```

constraint 内;

End of If

End of If

End of While

End of Procedure

循环执行以上这些操作,直至存放待处理特征线的集合为空时止.可用伪码表示为:

Procedure Constraint Delaunay Triangulation (DT, constraint)

//DT 是原始点集无约束条件下的 Delaunay 三角剖分结果;

// constraint 为全体待处理特征线的集合;

令 constraint-finish 为空集合;

While Not Empty (constraint)

从 constraint 中取出一条特征线 l;

If l 不包括在 DT 中 then

按第 2 节所述的方法在 l 上添加附加点,并对影响域重新进行三角剖分;

将被分割成的短特征线段放入 constraint-finish 中;

将影响域的三角剖分结果放入 T 中;

Edge-swap-triangle (T, constraint, constraint-finish);

End of If

End of While

End of Procedure

在这段伪码中,虽然为了描述方便,将待处理特征线的集合作为一个函数的参数,但事实上,额外新的特征线均可以在该函数的运行过程中随时加入.

整个处理过程可由图 3 所示的流程图表示.

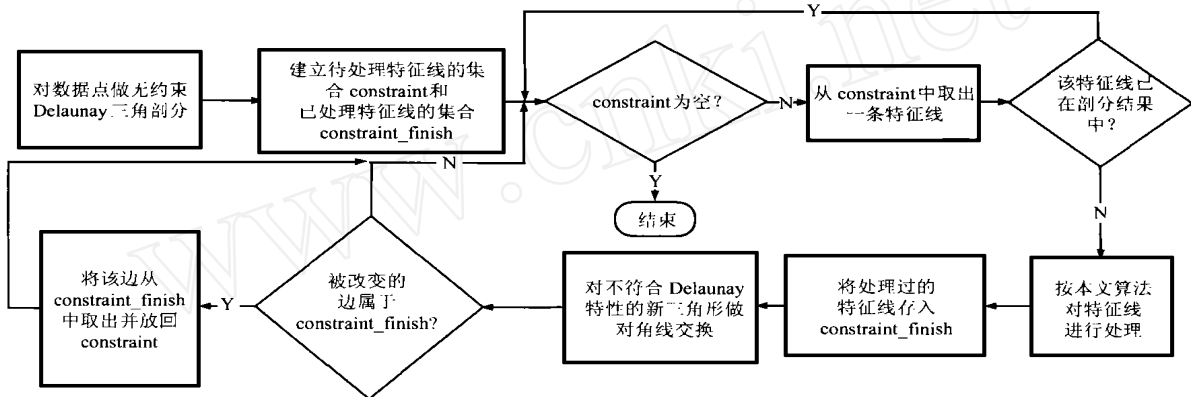


图 3 带特征约束的 Delaunay 三角剖分流程图

### 4 实验结果及讨论

为了检验本文所提出的算法,对平面上由 13 个数据点组成的点集进行了实验.在第一组实验中,在数据点集中规定 4 条特征线.图 4(a)即为数据点集在无约束条件下的 Delaunay 三角剖分结果,其中 4 条特征线已用虚线表示.这 4 条特征线中,有一条已出现在剖分结果中,因此不用再对其做进一步的处理.图 4(b)为采用文献[7]的方法对这些数据的处理结果;图 4(c)为用本文所提出的算法的处理结果.从中不难看出,

两种算法都可以保留特征线,但在用本文算法所得到的结果中,附加点的个数为 6;而在用文献[7]算法的结果中,附加点的个数为 14.在第二组实验中,特征线的数目增加为 7 条,其分布如图 5(a)所示.图 5(b)为采用文献[7]所得到的剖分结果,其中附加点的个数为 19.图 5(c)为用本文算法的结果,其中附加点的个数为 12.经检验,这两组实验的剖分结果均满足 Delaunay 特性.并且从这两组实验可以看出,采用本文的算法可以大大减少所必须的附加点数.

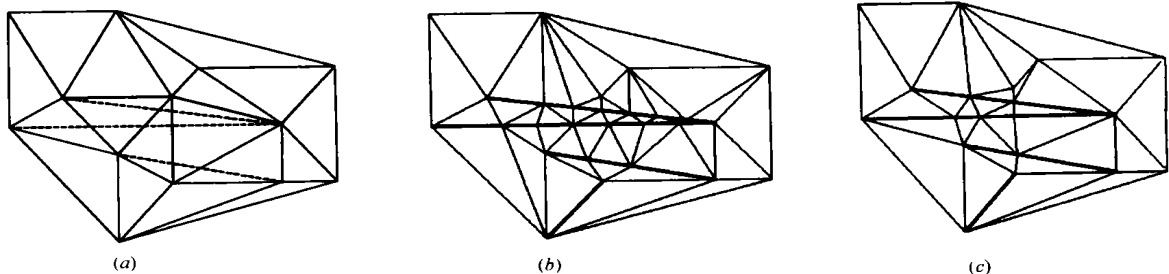


图 4 四条特征线时的实验结果.(a)无约束条件下的剖分结果;(b)用文献[7]算法得到的结果;(c)用本文算法得到的结果.

下面,再采用文献[6]中的数据分别用文献[6],文献[7]以及本文所提出的方法进行 Delaunay 三角剖分.图 6(a)为原

始的散乱数据,其中的直线为需要在剖分结果中保留的特征线.图 6(b)为采用文献[6]中的算法所得到的三角剖分结果,

图 6(c) 为用文献[7]中的算法而得到的剖分结果, 而图 6(d) 为采用本文方法得到的结果. 从中不难看出, 文献[6]所要增加的附加点的个数为 12, 文献[7]所要增加的附加点的个数

为 10, 而若采用本文所提出的算法, 则一共只需 6 个附加点即可.

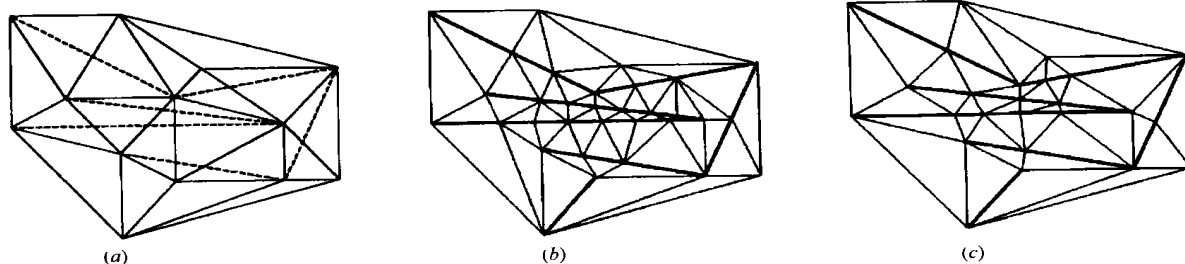


图 5 七条特征线时的实验结果. (a) 无约束条件下的剖分结果; (b) 用文献[7]算法得到的结果; (c) 用本文算法得到的结果.

## 5 结束语

本文提出了一种新的带特征约束条件的 Delaunay 三角剖分算法. 该算法通过在特征线上加入数据点而使得最终的剖分结果仍满足全局 Delaunay 特性. 较之于其它同类算法, 本算法更充分地利用了 Delaunay 三角剖分的圆准则, 因而所需要的附加点的个数可以大大减少. 并且, 因为本算法对所有特征线的处理是逐条进行的, 所以允许这些特征线在处理过程中逐条加入. 最终的实验结果证实了本算法的优越之处. 需要指出的是, 虽然采用本算法可以有效地减少附加点的数量, 但却不能保证所得到的结果是最优的. 至于小最应加入的点的数目和位置理论最优解为何, 还有待进一步研究.

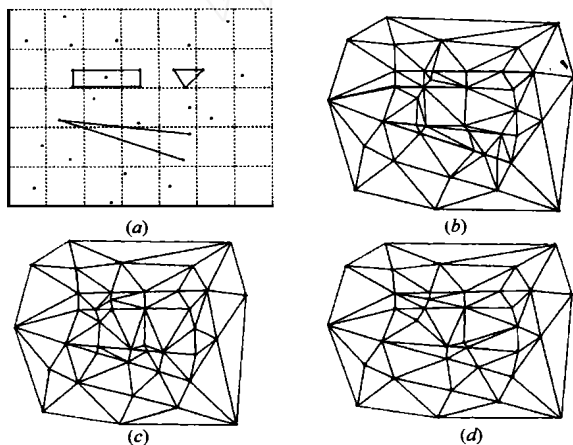


图 6 用文献[6]中的数据所得到的结果. (a) 为原始数据; (b) 为文献[6]中的实验结果; (c) 为用文献[7]的算法所得的结果; (d) 为用本文算法所得的结果

## 参考文献:

[ 1 ] Preparata, F P, Shamos, M I. Computational Geometry-an Introduction [M]. Springer, New York, Berlin, Heidelberg, 1985.

[ 2 ] Edelsbrunner, H. Algorithms in Combinatorial Geometry [M]. Springer, New York, 1987.

[ 3 ] De Floriani, L. An On-Line Algorithm for Constrained Delaunay Triangulation [J]. CVGIP: Graphical Model and Image Processing, 1992, 54(3): 290 - 300.

[ 4 ] Anglada, M V. An Improved Incremental Algorithm for Constructing Restricted Delaunay Triangulation [J]. Computer & Graphics, 1997, 21(2): 215 - 223.

[ 5 ] Faugeras, O. Three-Dimensional Computer Vision: a Geometric Viewpoint [M]. MIT press, Boston, 1993.

[ 6 ] Fang, T-P, Piegel, L A. Algorithm for Constrained Delaunay Triangulation [J]. Visual Computer, 1994, 10(5): 225 - 265.

[ 7 ] 卢朝阳, 吴成柯. 满足全局 Delaunay 特征的带特征约束的散乱数据最优三角剖分[J]. 计算机学报, 1997, 20(2): 118 - 124.

[ 8 ] De Floriani, L. A Pyramidal Data Structure for Triangle-based Surface Description [J]. IEEE Comput. Graphics Appl., 1989, 9(2): 67 - 68.

## 作者简介:



梅承力 男, 1974 年生于陕西渭南. 博士生. 分别于 1995 年和 1998 年获得西安电子科技大学信息工程系学士和硕士学位, 现在上海交大图像通信与信息处理研究所攻读博士学位. 主要研究兴趣为计算机视觉、图像处理、图像检索等.

肖高逾 男, 1973 年生于陕西西安. 博士, 主要研究兴趣为计算机视觉、图像处理等.

周源华 男, 1940 年生于江苏盐城. 博士生导师. 主要研究方向为计算机视觉、图像处理、多媒体技术等.