

平面多边形域的快速约束 Delaunay 三角化

曾 薇¹⁾ 孟祥旭¹⁾ 杨承磊¹⁾ 杨义军²⁾

¹⁾ { 山东大学计算机科学与技术学院 济南 250061 }

²⁾ { 清华大学计算机科学与技术系 北京 100084 }

(zengw@mail.sdu.edu.cn)

摘 要 针对任意平面多边形域,采用增量思想和均匀网格,在局部范围内快速生成约束 Delaunay 三角形.该方法不会生成区域外的三角形,对存在折线、离散点以及含“洞”的情况不需要特殊处理.实验结果表明,该方法对于随机生成的简单多边形域三角化速度快,平均计算时间呈近似线性.另外,针对文字、工业图案等带状图像的边界多边形,充分利用其近似等宽性优化算法,将其应用于带状图像骨架的快速提取.

关键词 平面多边形域;约束 Delaunay 三角化;均匀网格

中图法分类号 TP391

Fast Constrained Delaunay Triangulation for Planar Polygonal Domains

Zeng Wei¹⁾ Meng Xiangxu¹⁾ Yang Chenglei¹⁾ Yang Yijun²⁾

¹⁾ { School of Computer Science and Technology, Shandong University, Jinan 250061 }

²⁾ { Department of Computer Science and Technology, Tsinghua University, Beijing 100084 }

Abstract Based on the incremental idea and the uniform grid, constrained Delaunay triangles can be computed in local ranges for arbitrary planar polygonal domains. No triangles outside the valid region of the domain are computed and for domains with polylines, scattered points and holes, no extra operations are specially required. The tested analysis shows that for simple polygonal domains randomly generated, the algorithm is efficient in computation and has an almost linear run time. Furthermore, the algorithm is optimized on a special kind of polygons formed by the boundary of such band-images as texts, industrial patterns etc., with their characteristic of approximately equal width fully utilized. The feature has been applied for efficient skeleton extraction of band-like images.

Key words planar polygonal domains; constrained Delaunay triangulation (CDT); uniform grid

1 引 言

三角化是计算几何的重要研究内容之一,在曲面可视化裁剪、立体平版印刷技术、服装设计、地形建模、图形重建、曲面插值、有限元分析等领域中有着广泛的应用.计算几何学者已经研究了大量这一形式的问题:针对平面离散点集或平面区域及其优化标准,如何构建有效的最优三角化算法. Delaunay 三角化(Delaunay Triangulation, DT)是重要思想之

一.早期的关于 DT 的研究主要集中在平面离散点集上,其结果具有最大空外接圆^[1]和最小角最大^[2]两个非常重要的性质;并且满足空圆准则——排除 4 点共圆情况.一个三角形出现在 DT 中,当且仅当它的外接圆中不包含其他点,称这样的三角形是 Delaunay 的.在离散点之间添加边可将离散点集扩展为平面直线图(Planar Straight Line Graph, PSLG),形成开放或封闭的平面区域.若没有说明,文中提到的区域均指封闭区域,又称平面多边形;其 Delaunay 三角化有着较为广泛的应用.

一般地,平面多边形域 Ω 具有一条外边界和 h ($h \geq 0$) 条内边界,且内外边界分别为顺逆时针方向,区域位于内外边界边的左侧,如图 1a 所示;区域内可存在离散点或折线段,如图 1b 所示;内边界、外边界、折线段三者之间可有相同顶点,如图 1c 所示. 边界边和区域内的线段统称为约束边. 若 Ω 的两个顶点 p_1 和 p_2 构成的开线段 p_1p_2 (不含 p_1 和 p_2) 在 Ω 的内部,且不与 Ω 的任何约束边相交,则 p_1 与 p_2 是彼此可见的,将称线段 p_1p_2 为内部边. 对于约束边或内部边 p_1p_2 及顶点 p ,若 p 与 p_1 、 p 与 p_2 均彼此可见,则 p 是 p_1p_2 的一个可见点.

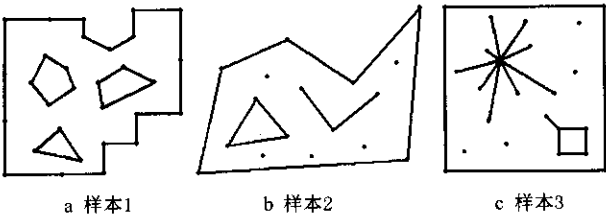


图 1 平面多边形域

如图 2 所示,对平面多边形域的约束 DT (Constrained Delaunay Triangulation, CDT) 不同于离散点集 DT,其结果能够覆盖所有约束边和顶点;同时,仍保留了 DT 最小角最大等特性. 它满足约束 Delaunay 标准,即每个三角形 t 的外接圆内均不存在相对于 t 的任何一条边可见的多边形域的顶点,并且 t 不与约束边相交(可见性只与约束边有关). 这样的三角形称为约束 Delaunay 三角形,其中与一条边相对的顶点称为该边的 DT 点.

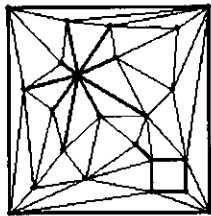


图 2 图 1c 的 DT

针对平面离散点集, Bentley 等^[3]证明,对于一致分布的 2D 点集,借助均匀网格从一个点搜索其最近点的操作可在常数期望时间内完成;并且给出了最优期望时间为 $O(n)$ 的 2D 点集(点数为 n) Voronoi 图生成算法. DT 是 Voronoi 图的对偶图^[4,5]. Fang 等^[6]基于网格实现了离散点集的 DT,将 DT 点的搜索限制在局部范围内,速度较快;并对随机点集测试表明,时间复杂度呈近似线性.

Piegl 等^[7]将文献[6]思想扩展到平面多边形域.在为边寻找顶点的过程中考虑约束边的存在,使生成的三角形满足约束 Delaunay 标准.该算法依据边界边的凸凹性和对三角化的影响来设计输出结构,过滤掉多余的外部三角形(多边形域外的三角形).因此,它需要消耗大量判断处理时间,特别是对于图 3 所示的工程图、指纹、文字、工业图案等带状图像的边界多边形域更是如此.李伟青等^[8]提出,将平面多边形域的所有顶点按扫描方式排列,按有序点寻找局部区域的一条边,构造新的三角形.该算法经调整也能够处理平面离散点集,但是为满足设定的优化标准,需要进行边翻转等操作.

针对内部不含离散点和折线的平面多边形域, Klein^[9]基于网格和分而治之思想实现其 CDT.该算法选择一条“合适”的边界边,按文献[7]算法构造约束 Delaunay 三角形,将多边形等分为两个子多边形,并按此方法处理子多边形.算法简洁,速度较快,但是,需要预先将带“洞”的多边形转换为不带“洞”的情况后再进行处理.另外,分而治之的方法本身会产生大量的负载开销,而且选择用于分割的“合适”的边界边也是比较困难的.

在上述算法基础上,本文给出一种适用于任意平面多边形域的快速 CDT 算法,并充分利用文字、指纹、工程图、工业图案等带状图像的边界形成的一类多边形——带状多边形的特性优化其 CDT.将初始边的选择稍作修改,本文算法也可用于计算平面离散点集或任意 PSLG 的 DT.



图 3 带状图像

2 算法思想

本文算法利用网格数据结构来加速整个平面多边形域的 CDT. 图 4 所示为剖分过程示意图,其虚线表示已生成但未处理的内部边,存放在堆栈中,粗实线表示当前处理的边. 算法具体步骤如下:

Step1. 建立单元大小为 $E \times E$ 的均匀网格,并将多边形域的顶点和边放入其中. 其中 $E = \sqrt{(b_w \times b_h)/n}$, b_w 和 b_h

分别为多边形域包围盒的宽度、高度, n 为多边形域的顶点数.

Step2. 取任意一条外边界边 $p_1 p_2$.

Step3. 计算 DT 点 p_3 , 构成约束 Delaunay 三角形 $\triangle p_1 p_2 p_3$.

Step4. 如果新生成的边 $p_1 p_3$ 不是约束边,若已经在堆栈中,则将其从中删除,否则,将其放入堆栈. 类似地,可处理 $p_3 p_2$.

Step5. 若堆栈不空,则从中取出一条边,转 Step3, 否则,算法停止.

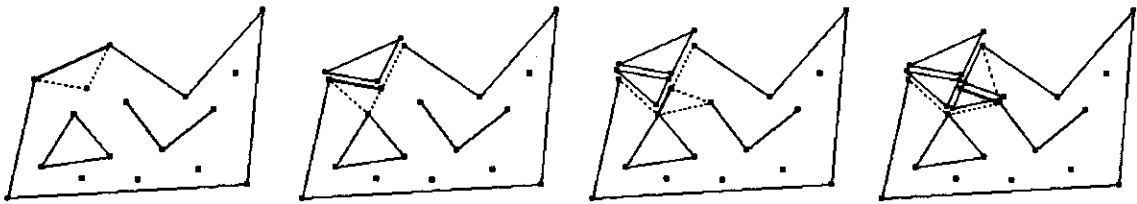


图 4 三角化过程

3 创建网格结构

在网格结构中,各网格单元 b 必须对应一个 b 内顶点链表 $Point(b)$ 和一个经过 b 的边链表 $Edge(b)$. 整个网格结构使用二维数组来表示. 若将一个顶点 $p(x, y)$ 放入相应的网格单元,其对应的数组下标为 $(\lceil x - x_{min}/E \rceil, \lceil y - y_{min}/E \rceil)$, 这里 (x_{min}, y_{min}) 是包围盒最小点,若将多边形域中的一条边界边放入其经过的网格单元,其对应的数组下标可通过 RayTracing 技术^[10]快速计算得到.

整个平面多边形域使用一个顶点链表 $polygon$ 来记录,约束边由顶点间的连接关系隐含表示. 依次记录外边界、内边界、折线、离散点(三者顺序可调);各边界和折线上的顶点按约定顺序排列. 如图 5、6 所示,整个域信息由头指针 $head$ 串起,其中的

对象 $polygon1$, $polygon2$, $line3$, $point4$ 以及 $point5$ 在存储的同一层次;各个顶点的真实信息仅存在于 $polygon$ 各节点中;网格单元的顶点链表节点和边链表节点都通过指针引用 $polygon$ 节点获得顶点或相关联的边信息,这样,既节省空间又能够保证数据一致.

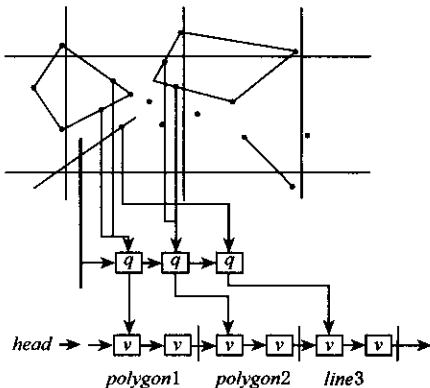


图 6 网格单元边链表

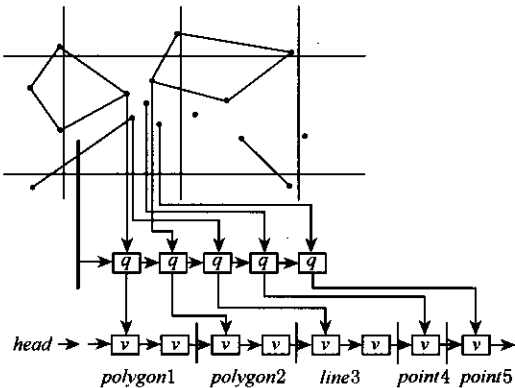


图 5 网格单元顶点链表

4 计算 DT 点

将计算一条边的 DT 点的过程称为一趟. 每趟包含两步: 寻找初始可见点和确定 DT 点. 过程中借助均匀网格,将 DT 点的搜索限制在局部范围内,加快计算速度.

4.1 寻找初始可见点

如图 7 所示,假定当前需要处理的边为 $p_1 p_2$. 那么以 $p_1 p_2$ 的中点 p_{mid} 所在的网格单元为中心,

按照螺旋式扩散的顺序搜索网格单元,对其顶点链表中各节点作相对 p_1p_2 的可见性判断,直至找到一个可见点 p_3 ,则称其为初始可见点.每扫描一个网格单元,就将其标记为当前趟数.当网格单元中包含多个 p_1p_2 的可见点 $p_i(i > 1)$ 时,选取使得 $\angle p_1p_ip_2$ 最大的可见点.螺旋式搜索在一定程度上保证在 p_1p_2 近处找到可见点,从而缩小搜索最优可见点的范围,但这不是绝对的.因此,初始可见点不是惟一的,具有一定的搜索随机性.

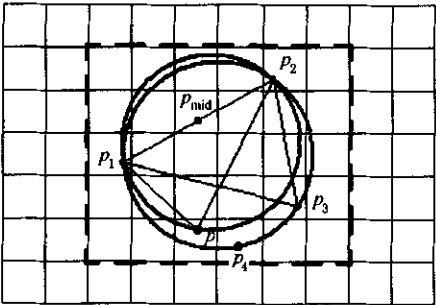


图7 计算DT点

我们称 p_3 为 p_1p_2 的可见点,其必须满足下面三个条件:

- (1) p_3 在边 p_1p_2 的左侧;
- (2) p_3 与 p_1 可见,即对于 p_1p_3 所穿过的所有网格单元中记录的任意一条边 e ,有 $e \cap (p_1p_3) = \emptyset$;
- (3) p_3 与 p_2 可见.

4.2 确定DT点

对于均匀网格上的任意一条线段 p_1p_2 ,将其最小包围盒 $B_m(p_1p_2)$ 内(包含相交)的网格单元构成的包围盒称为 p_1p_2 的网格包围盒.

确定DT点的过程如下:

- Step1. 构造 $\triangle p_1p_2p_3$ 的外接圆 $C(p_1, p_2, p_3)$ 及其网格包围盒 $B(C(p_1, p_2, p_3))$ (如图7虚线所示).
- Step2. 依次访问网格包围盒内的每个网格单元:
- 对未作当前趟数标记的网格单元进行搜索,并将其标记为当前趟数.若某个网格单元中存在可见点 p 并且 $\angle p_1pp_2 > \angle p_1p_3p_2$ 则令 $p_3 = p$ 转 Step1,否则,转 Step3.

Step3. 若当前网格包围盒内所有网格单元都已被标记为当前趟数,也即 $C(p_1, p_2, p_3)$ 内无可见点,则 p_3 为 p_1p_2 的DT点.

当相对 p_1p_2 恰有两个可见点 p_3 和 p_4 并满足 $\angle p_1p_3p_2 = \angle p_1p_4p_2$,即4点共圆时,在共圆点 p_3 被选作DT点后,4点共圆不复存在;新生成的Delaunay三角形内部边 p_1p_3 或 p_3p_2 中必有一条与 p_4 可见的内部边,它必然选中 p_4 作为DT点,从而实现4点共圆下的CDT.可见,共圆可见点中的一个一旦被选作DT点,其余的对此次搜索不再起作用,正是因为计算的不重复性以及增量式生长结构的前进性,避免了死循环,自动有效地解决了4点共圆的情况.图8所示多点共圆的情况与此类似,其中 (c, s, h, t) 表示(约束边数,离散点数,内边界数,折线段数).另外,通过动态更新包围盒,可以有效地收缩可见点范围,并充分利用网格单元搜索标记,避免重复操作以加快速度.

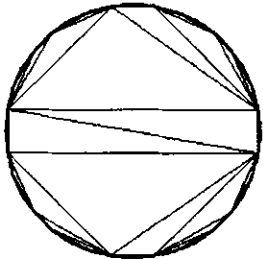


图8 CDT实例 $(c, s, h, t) = (36, 0, 0, 0)$

5 带状多边形的处理

骨架能够有效地标识如图3和图9a所示文字、指纹、工程图、工业图案等带状图像特征.对这类图像骨架化是模式识别和计算机视觉等领域中的一个十分重要的内容.通过对其边界多边形(如图9b所示)进行CDT(如图9c所示)根据三角形的类型计算带状图像的骨架(如图9d所示)是一种较快的方法^[11].同时,由于三角形按生长方式存储,离散的骨架具有某种程度上的有序连接性.

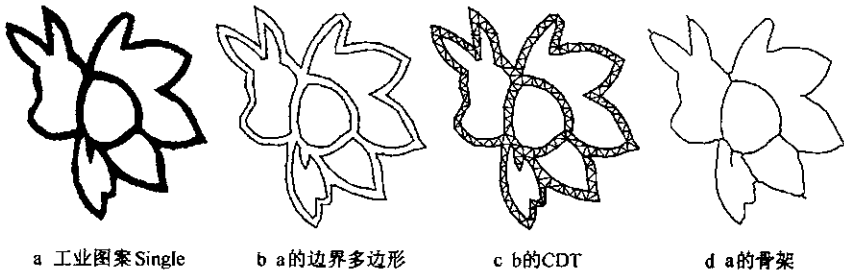


图9 带状图像及其骨架计算

如图 9 ,对于一幅带状图像 ,设宽度近似为 D . 对其边界多边形矢量化 ,使每边长度 $E \leq 2D$,可得到如图 9b 所示带状多边形 . 显然 ,它是平面多边形域的特例 ,信息存储类似 . 充分利用其特点 ,对网格结构的创建方法加以修改 ,来提高计算速度及节约空间 . 具体地 ,在创建网格时 ,将带状多边形的包围盒向四周扩展 ,使其长度和高度均为 E 的倍数 ,同时所有顶点不会落在包围盒边界上 . 由于带状多边形边长的均等性以及网格宽度与边长的约束关系 ,与一条线段相交的边界边可在局部范围内根据边界边的顶点找到 ,因此每个网格单元只保存位于其内的多边形顶点信息即可 .

实验表明 ,基于此网格结构可快速实现带状多边形的 CDT . 当顶点满足一致分布并且边的长度及带状图像的宽度为 $O(E)$ 时 ,可证明 CDT 算法时间复杂性为线性的^[3] .

另外 ,将初始边的选择方法稍作修改 ,即可用于平面离散点集或任意 PSLG 的 DT .

6 实验结果及分析

6.1 实 例

图 10~15 给出了一些实例 ,算法处理对象为包含离散点和折线的平面多边形域 . 图 10 中给出一个规则图案 ,外边界和内边界均线段化 . 图 11 给出了一件服装设计样本实例 . 图 12 中外边界内仅存在离散点 ,而图 13 是在其基础上添加 2 条内边界和 1 条桥 . 图 8 是正 36 多边形逼近的圆的 CDT 结果 . 图 14 ,15 分别是在图 8 的中心位置添加 1 条内边界和 1 个离散点的 CDT 结果 . 这表明 ,针对 4 点共圆 ,本文算法能够在一定的误差允许下快速实现 CDT . 此外 ,图 10 ,12~14 ,中均包含多点共线的情况 .

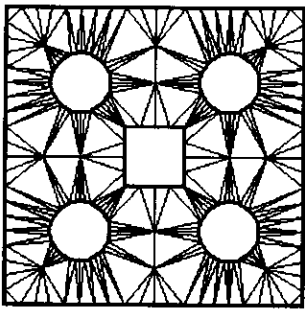


图 10 规则图案 $(c, s, h, it) = (240, 12, 5, 0)$

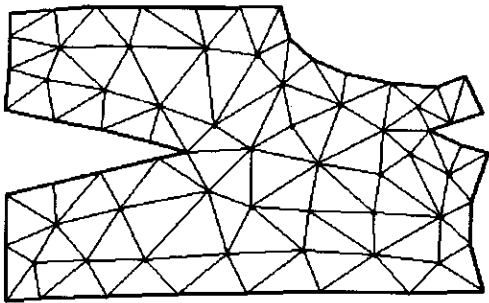


图 11 CDT 实例 $(c, s, h, it) = (40, 32, 0, 0)$

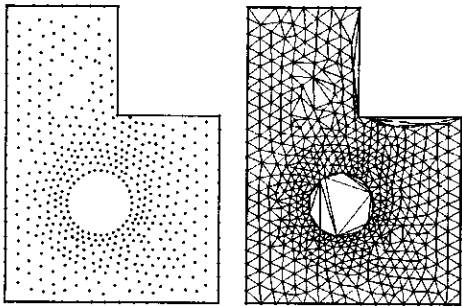


图 12 CDT 实例 $(c, s, h, it) = (73, 523, 0, 0)$

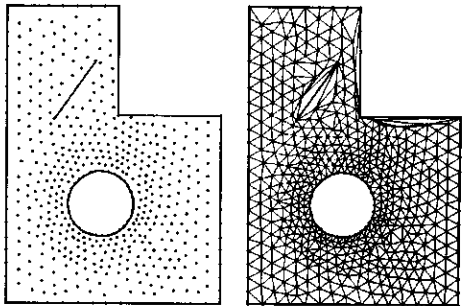


图 13 CDT 实例 $(c, s, h, it) = (128, 467, 2, 1)$

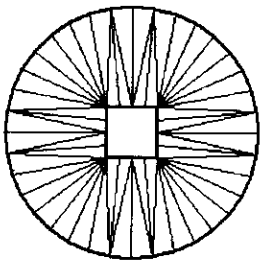


图 14 CDT 实例 $(c, s, h, it) = (52, 0, 1, 0)$

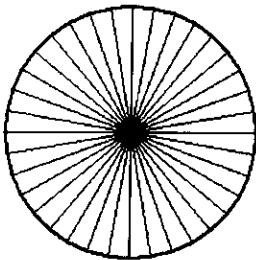


图 15 CDT 实例 $(c, s, h, it) = (36, 1, 0, 0)$

6.2 参数分析

一般情况下,基于网格的平面多边形区域的 CDT 算法的运行时间主要消耗在点的可见性判断上. 在本文中,可见性只与约束边相关. 通常在点数固定不变的情况下,通过连接两离散点构造约束边得到如图 16 所示的多边形域,CDT 运算时间随着约束边边数和约束边边长的增加而增加.

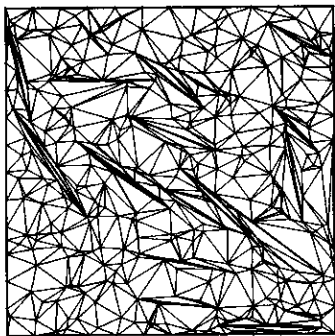


图 16 平面多边形域 CDT (c s h t)=(78 300 0 30)

特别地,对于指纹、手写体文字以及大部分工业图案的带状多边形,实际测试表明,网格宽度 E 在 $2D$ 的附近取值时 CDT 时间消耗最少,如图 17 所示.但是,对于工程图,由于其边界一般多为较长的直线, E 在 $2D$ 的附近取值时 CDT 时间消耗不一定最少. 因此,用户可根据实际情况设置参数 E ,且范围不超过该多边形包围盒宽度 b_w ;否则,整个网格仅包含一个单元,网格的采用就失去意义. 而且,若 $E > \sqrt{2} b_w$,则 CDT 时间不再改变,因为这时由给定带状图像的带状多边形(顶点数和边长)不再改变.

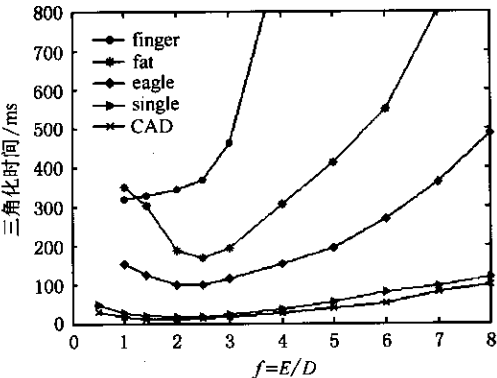


图 17 图 3a—e 和图 9a 中 5 幅图像的带状多边形 CDT 时间与 $f = E/D$ 的关系

6.3 时间比较

如图 18 所示,针对在外边界范围限定为 2000×2000 时随机生成的具有不同顶点数(约束边数)的

简单多边形域(不含离散点、折线、内边界),从表 1、图 19 可看到,本文算法时间呈近似线性,且明显快于 Klein 算法. 如图 20 所示,针对随机生成的具有不同顶点数的带状简单多边形域,表 2 给出了在 $E=2D$ 时本文算法与 Klein 算法的 CDT 运算时间,图 21 表明,两种算法都呈近似线性,但本文算法快于 Klein 算法. 以上时间估计均为在同一顶点数下取 100 个随机多边形域,求其 CDT 时间的平均值.

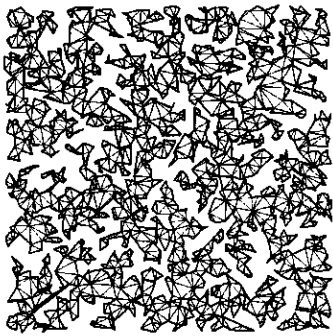


图 18 随机简单多边形域 CDT (c s h t)=(2000 0 0 0)

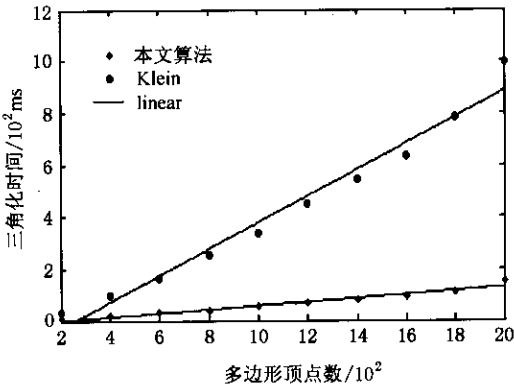


图 19 本文算法与 Klein 算法对随机简单多边形域的 CDT 时间比较

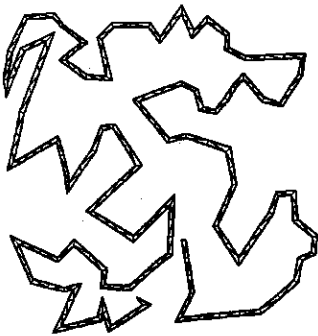


图 20 随机带状简单多边形域 CDT (c s h t)=(282 0 0 0)

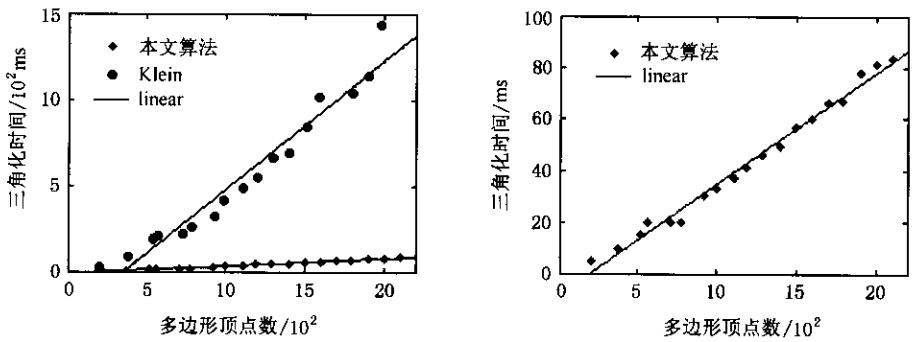


图 21 本文算法与 Klein 算法对随机简单带状多边形域的 CDT 时间比较

表 1 本文算法与 Klein 算法对随机简单多边形域的运行时间 ms

顶点数	200	400	600	800	1000	1200	1400	1600	1800	2000
Klein 算法	34	96	165	251	335	448	541	631	780	998
本文算法	8	18	30	40	55	66	80	95	111	150

表 2 本文算法与 Klein 算法对带状简单多边形域的运行时间 ms

顶点数	200	400	600	800	1000	1200	1400	1600	1800	2000
Klein 算法	29	91	212	270	422	550	692	1020	1138	1470
本文算法	6	12	19	28	33	41	49	60	67	81

本文算法与 Klein 算法均基于同一软硬件平台,使用 Microsoft Visual C++ 6.0 实现;针对同一数据集,在 CPU 主频为 1.9GHz、内存 256MB 的 PC 及 Microsoft Windows XP 操作系统上采集运算时间,并且,Klein 算法采用其最优方法 fl^[7]来分割多边形.两算法均基于网格来查找 DT 点,搜索思想一致,但 Klein 算法需要进行大量的分割边选择和网格结构重建,而本文算法过程中仅需对堆栈进行维护,两者对整体数据集的操作有较大差别,因此速度有明显区别.同时,针对平面简单多边形域,两算法均为 CDT;另外,排除多点共圆,平面多边形域的 CDT 是惟一的^[7],若存在共圆可见点,Klein 算法生成的 CDT 也有一定的随机性.

7 结 论

本文提出了一个快速、简单的平面多边形域的 CDT 算法.算法基于均匀网格,能够在其邻近区域内快速找到一条边的 DT 点.对于文字、指纹、工程图、工业图案等带状图像的边界多边形,算法可充分利用网格和多边形的特点进行快速 CDT.基于此的带状图像骨架化算法在商品化软件“集成化计算机辅助图案设计制版系统”中用于工业图案的修复(专业术语为“细茎重描”),可极大地减少设计人员

的工作量.

将初始边的选择方法稍作修改,即可用于计算平面离散点集或平面直线图的 DT.与文献 7 算法相比,本文算法不会生成区域外的三角形,并且对带“洞”区域和区域内存在折线、离散点的情况不需要额外操作.实验表明,对于随机(带状)简单多边形域,本文算法运行时间均近似呈线性,并明显快于 Klein 算法^[9].同时,算法能够有效地处理顶点、约束边的重叠情况和多点共线、4 点共圆等退化情况,从而能够稳定、正确地处理各种域信息,并且其效率满足应用的需求,因此具有更广泛的实际应用价值.

参 考 文 献

[1] Azevedo E F D, Simpson R B. On optimal interpolation triangle incidences [J]. SIAM Journal on Scientific and Statistical Computing, 1989, 10(6): 1063~1075

[2] Sibson R. Locally equiangular triangulations[J]. The Computer Journal, 1978, 21(3): 243~245

[3] Bentley J L, Weide B W, Yao A C. Optimal expected-time algorithms for closest point problems[J]. ACM Transactions on Mathematical Software(TOMS), 1980, 6(4): 563~580

[4] Gold C M. Three approaches to automated topology and how computational geometry helps[A]. In : Proceedings of the 6th International Symposium on Spatial Data Handling , Edinburgh , 1994. 145~158

- [5] Sack J R, Urrutia J. Handbook of Computational Geometry [M]. Amsterdam: Elsevier Science, 2000
- [6] Fang Tsung Pao, Piegl Les A. Delaunay triangulation using a uniform grid[J]. IEEE Computer Graphics and Applications, 1993, 13(3): 36~47
- [7] Piegl L A, Richard A M. Algorithm and data structure for triangulating multiply connected polygonal domains [J]. Computer and Graphics, 1993, 17(5): 563~574
- [8] Li Weiqing, Peng Qunsheng. A general and fast triangulation algorithm[J]. Chinese Journal of Computer-Aided Design and Computer Graphics, 2001, 13(9): 769~773 (In Chinese)
(李伟青, 彭群生. 一个通用的快速三角化算法[J]. 计算机辅助设计与图形学学报, 2001, 13(9): 769~773)
- [9] Klein Reinhard. Construction of the Constrained Delaunay Triangulation of a Polygonal Domain[M]. In CAD—Tools for Products. New York: Springer Verlag, 1996
- [10] Amanatides John, Woo Andrew. A fast voxel traversal algorithm for ray tracing [A]. In: Proceedings of Eurographics '87 Conference, Amsterdam, 1987. 3~10
- [11] Yang Yijun, Meng Xiangxu, Yang Chenglei, et al. Fast triangulation and thinning algorithm for complex band images [J]. Chinese Journal of Computer-Aided Design and Computer Graphics, 2003, 15(10): 1270~1274 (in Chinese)
(杨义军, 孟祥旭, 杨承磊, 等. 复杂带状图像的快速三角剖分与骨架化算法[J]. 计算机辅助设计与图形学学报, 2003, 15(10): 1270~1274)



曾 薇 女, 1980 年生, 硕士研究生, 主要研究方向为计算几何、图像处理、模式识别.



孟祥旭 男, 1962 年生, 博士, 教授、博士生导师, 主要研究方向为人机交互与虚拟现实、计算机图形学.



杨承磊 男, 1972 年生, 博士, 副教授, 主要研究方向为人交互与虚拟现实、图像处理、计算几何.



杨义军 男, 1979 年生, 博士研究生, 主要研究方向为 CAD/CAM、计算机图形学、计算几何.