第 26 卷 第 2 期 华侨大学学报(自然科学版) Vol. 26 No. 2
2005 年 4 月 Journal of Huaqiao University (Natural Science) Apr. 2005

文章编号 1000-5013(2005)02-0199-04

三角网自动连接的聚焦算法

王 华 军

(华侨大学信息科学与工程学院,福建 泉州 362021)

摘要 在三角网生长法的基础上,采用面向对象的技术,利用点数组和点索引数组来存贮平面上的散乱数据点. 基于 Delaunay 三角剖分的"圆准则",提出三角网自动连接的聚焦算法. 该算法在扩展新三角形时,将点的搜索范围控制在已知三角形的外接圆内,计算速度大大加快. 从给出的算例表明,该算法十分有效,特别适合于大数据量的三角剖分.

关键词 Delaunay, 三角剖分, 等值线, 计算几何

中图分类号 P 207; TP 391.41 **文献标识码** A

平面上散乱数据点的 Delaunay 三角剖分,在等值线图绘制、有限元分析、数字地面模型(DEM)和计算机视觉等领域有重要应用[1]. 国内外许多学者对其进行了广泛深入的研究[2]. Tsai 根据实现过程,把 Delaunay 三角剖分的各种算法归为 3 类,即分治算法、逐点插入法和三角网生长法[3]. 传统的三角网连接算法[4,5]在扩展新三角形时,对点的查找过程需对全平面上的点进行搜索和比较. 为此,它仅适合于散乱数据点较少的情况;而当数据量较大时,效率较低,计算速度较慢. 为了解决这一问题,凌海滨等[6]引进了封闭点的概念,将对生成新三角形不再有贡献的点删除,加速了算法;毛善君等[7]提出了从相邻有限矩形网格中,寻找当前扩展边的扩展点的算法. 本文采用面向对象的技术,利用点数组和点索引数组来存贮平面上的散乱数据点. 基于 Delaunay 三角剖分的"圆准则",提出了一种新的算法. 该算法在扩展新三角形时,将点的搜索范围控制在已知三角形的外接圆内,计算速度大大加快.

1 基本概念

定义 1 域 D 中的散乱点集 S,求其 Delaunay 三角剖分. 它们满足如下 3 点剖分条件. (1) 尽可能避免出现具有尖锐内角且形状狭长的三角形,也即三角形三边尽可能相等或三内角尽可能相等. (2) 三角形之间互不交叉. (3) 三角形不发生重复. 满足剖分条件的三角形称为合法三角形.

定义 2 当由 $\triangle ABC$ 的 AB 边扩展生成新 $\triangle ABD$ 时,称 C 点为参考点,AB 为扩展边,D 点为扩展点,并记 $D = \mathrm{Expand}(A, B, C)$.

定义 3 若三角形 T 的顶点中包含点 P,则称 T 与 P 相关联,记为 $R(P, T)$. 显然,点 P 可与多个三角形 T_i 相关联,记作 $R(P, T_0, T_1, \cdots, T_k)$. 其中 $T_0, T_1, \cdots, T_k$ 绕 P 按逆时针方向排列,如图 1 所示. 称 T_0 为与 P 相关联的头三角形,记 $T_h(P) = T_0$. 相应地,称 T_k 为与 P 相关联的尾三角形,记 $T_t(P) = T_k$. 称 $T_h(P)$ 中按逆时针方向继 P 点后的第 1 顶点 V_0 为 P 的头点,记为 $\mathrm{Head}(P)$. 相应地,称 $T_t(P)$ 中按顺时针方向继 P 点后的第 1 顶点 V_{k+1} 为 P 的尾点,记为 $\mathrm{Tail}(P)$.

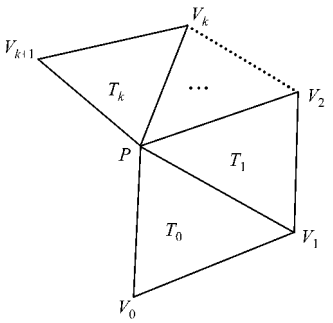

图 1 与点 P 相关联的三角形

定义 4 点 P 能否与其它点 V 构成合法三角形的标志,称为点 P 的扩展状态 $\mathrm{Flag}(P)$. 能则称为 TRUE;否则,为 FALSE. 其初值为 TRUE. 然后在扩展三角形时,动态

收稿日期 2004-10-19

作者简介 王华军(1972-),男,硕士研究生,主要从事电磁法及计算机应用的研究. E-mail:huajunw@163.com

基金项目 福建省青年科技人才创新基金资助项目(2001J007)

地改变其状态. 若与点 P 相关联的三角形均已求出,特别地,当这些三角形正好绕 P 点转一圈时,称 P 点为封闭点[6]. 它不可能再对扩展三角形有贡献. 此时,将 $F(P)$ 由 TRUE 变为 FALSE. 在以后的查找过程中,将略去该点,加速查找.

定义 5 用于存贮给定平面上的散乱点 Dot 的数组,称为点数组 DotArray. 其中,每个点都包含以下信息:点编号、点坐标(x, y)、与该点相关联的三角形、点扩展状态及其它.

定义 6 平面上的每个点都包含两个独立的坐标分量 x 和 y. 把其中 x 坐标相同的点归为一类,设共有 M 类,再把每类中的点,按 y 坐标的大小来排序,形成 Y 索引数组. 显然,共有 M 个 y 索引数组. 现将其按 x 坐标的大小来排序,便可生成点索引数组. 可见,点索引数组是由 M 个 x 坐标相同的 Y 索引数组组成的,其元素均有一个 x 坐标和一个指向 Y 索引数组的指针,而每个 Y 索引数组的元素又都包含一个 y 坐标和一个索引号. 即点(x, y)在点数组中的索引号,如图 2 所示. 图中,$x_0 < x_1 < \cdots < x_i < \cdots < x_M$,$y_0^i < y_1^i < \cdots < y_j^i < \cdots < y_N^i$. 进一步可知,任给定一索引点对$(i, j)$,则它唯一地与点数组中的一点 P 对应,记作 $P = \mathrm{DotIndex}(i, j)$.

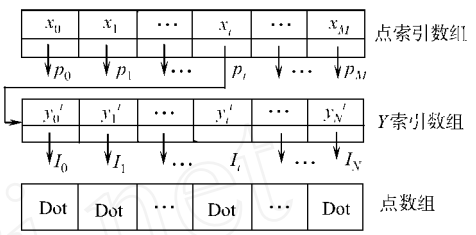

图 2 点索引数组示意图

定义 7 用于存贮生成三角形的数组,称为三角形数组 TriArray. 其中每个三角形都有其编号、顶点索引数组等信息.

2 算法原理与实现

常规 Delaunay 三角剖分是先生成一个满足剖分要求的三角形 $\triangle ABC$,然后由 $\triangle ABC$ 往外扩展并连接全部散乱点构成三角网. 本文的算法也基于这种思想,但有 2 点不同. (1) 在扩展三角形时,前者是以三角形的边为线索,本文则以三角形的顶点为线索. 它一次就将与该顶点有关的三角形全部扩展完毕(图 3). 这样的好处是每完成一个循环,都有点的扩展状态由 TRUE 变为 FALSE,而不参与以后的运算,可提高计算速度. (2) 在扩展三角形时,采用了聚焦算法,是整个算法中最关键的一步. 下面以 C 为参考点,AB 为扩展边来搜寻扩展点 D($D = \mathrm{Expand}(A, B, C)$)形成新三角形 $\triangle ABD$ 的过程为例(图 4),

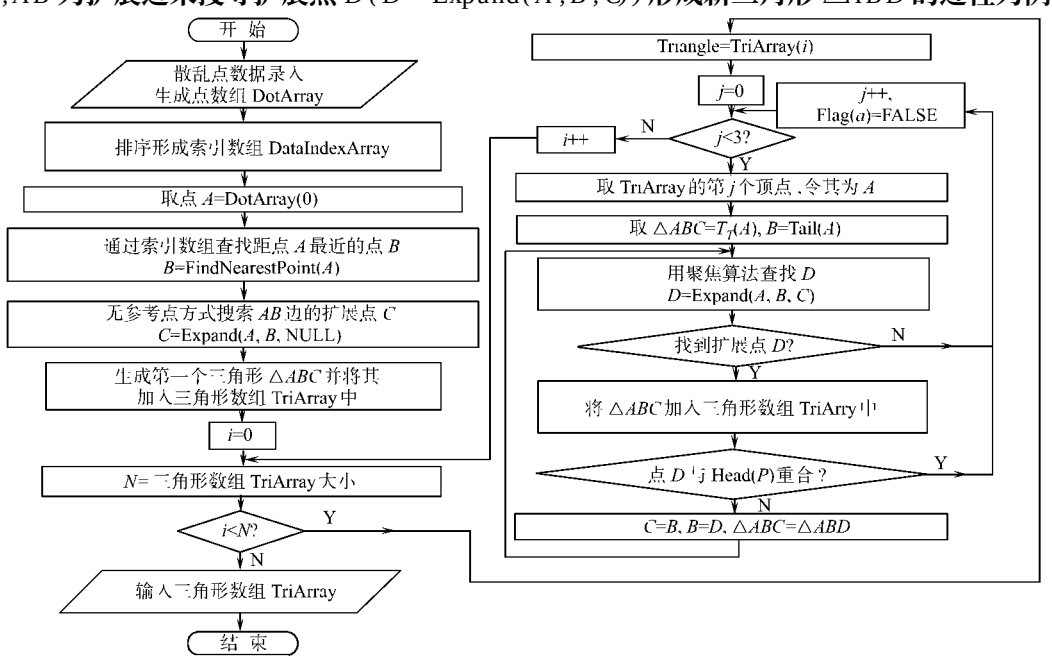

图 3 Delaunay 三角剖分程序流程图

程序流程如图 5 所示. 根据三角网自动连接的异号扩展准则[2],应在与点 C 分布于直线 AB 异侧的区域 Ω_0 内查找距 AB 中点 O_0 最近的点 D_0,作为初始扩展点. 搜索半径为 $R_0 = O_0D_0$,则可生成 $\triangle ABD_0$. 它不一定满足 Delaunay 剖分条件. 现在的问题,是否存在更好的扩展点. 作 $\triangle ABD_0$ 的外接圆,其圆心为

O_1,半径为 R_1. 由三角网自动连接的"圆准则"可知[2],比 D_0 更好的扩展点必位于圆 O_1 内. 考虑到刚才

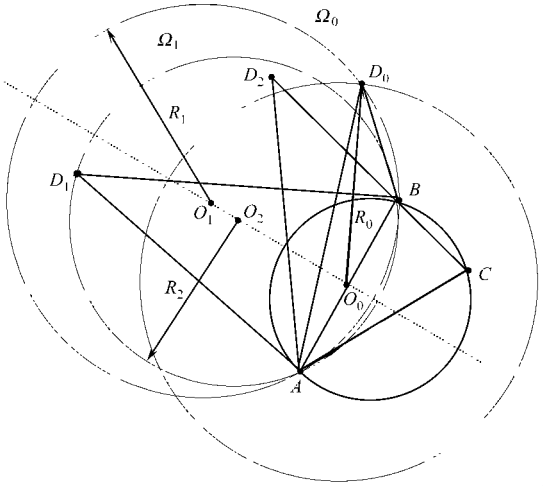

图 4　扩展点的外接圆逐步搜索算法示意图

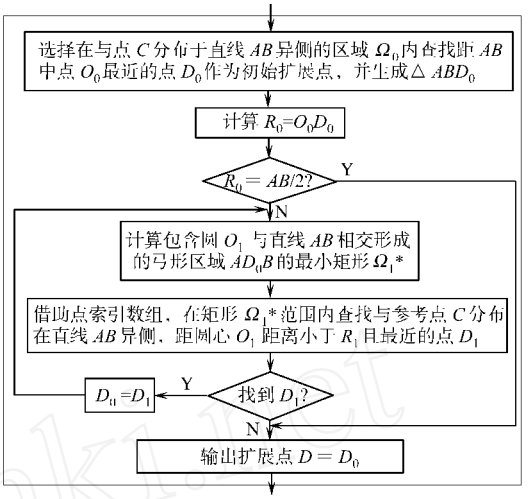

图 5　用聚焦算法查找扩展点 D 的程序框图

已搜索过的圆心在 O_0,半径为 R_0 的圆及其和圆 O_1 的位置关系,有如下两种情况. (1) 若 $R_0 = AB/2$,则圆 O_1 必与圆 O_0 重合. 显然,不可能有比 D_0 更好的扩展点. D 就是 D_0,搜索停止. (2) 若 $R_0 \neq AB/2$,则圆 O_1 必与圆 O_0 相交或相切. 此时,均需进一步搜索. 搜索范围已从半平面 Ω_0 的缩小到圆 O_1 以内,且与 C 在直线 AB 的异侧的弓形区域 Ω_1 内. 进一步看,要求限制在圆 O_1 内的点,也即到圆心 O_1 的距离小于圆的半径 R_1 的点. 并且,要做到扩展的三角形 3 边尽可能相等,或 3 内角尽可能相等,则 AB 中垂线上的点 O_1 将是最佳点. 因此,搜索过程变为在 Ω_1 内寻找距 O_1 最近点 D_1. 在程序设计时,为了方便,往往将 Ω_1 变为在包含该弓形区域的最小矩形 Ω_1^*. 借助点索引数组,不难实现在矩形 Ω_1^* 范围内查找与参考点 C 分布在直线 AB 异侧,距点 O_1 距离不超过 R_1 且最近的点 D_1. 重复这搜索过程,直至找不到更合适的点,最终将得到最佳扩展点 D,并由此求得新三角形 $\triangle ABD$.

为了比较效果,笔者用 VC^{++}6.0 实现了该算法,在赛扬 550, 128 M 内存的微机上,用程序产生的随机点列进行了计算. 结果与文[6]进行对比,如表 1 所示. 表中 M 为生成的三角形数目,t 为计算所需的时间. 由表可以看出,在排除计算机本身的速度差异外,本算法在计算速度上有成倍增长. 图 6 为 3 000 个随机点的计算实例(局部).

表 1　算法效率比较表

点数	文[6]算法		本文算法	
	M	t/ s	M	t/ s
3 000	5 382	7	5 980	1
5 000	8 239	18	9 975	3
10 000	14 573	58	19 977	10
20 000	23 074	151	39 964	32

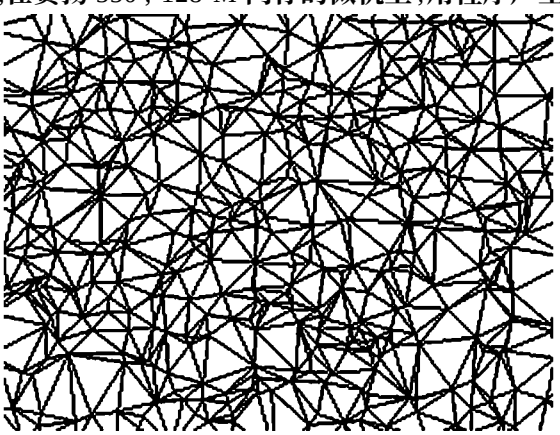

图 6　Delaunay 三角剖分实例(局部)

3　算法分析

由程序流程图可知,整个三角剖分的时间排序形成索引数组的时间,以及扩展三角形并连成三角网所需的时间由两部分组成.

设平面上的 n 个离散点,根据 x 坐标的不同可分为 M 类,每类有 L_i 个不同 y 坐标的点. 显然,有 $\Sigma L_i = n$,其均值为 $L = n/M$. 当用两次二分法在索引数组中查找平面上的某点时,其时间复杂度为

$$S_1 = O(\log_2 M + \log_2 L_i) \leq O(2\log_2 n) = O(\log_2 n).$$

采用二分插入排序形成索引数组,在最坏情况下,其时间复杂度为 $S_2 = O(n^2)$. 如果采用其它更有效的排序方法(如堆排序),则可使其时间复杂度降为 $S_2 = O(n\log_2 n)$.

在用聚焦算法扩展 AB 边时,由于在初始扩展点 D_0 求出后,搜索范围就缩小到 $\triangle ABD_0$ 的外接圆内,且与参考点 C 分布于 AB 边的异侧的弓形区域 Ω 内. 此时,已与平面上的总点数 n 没有关系,只与域 Ω 内的点数 q 有关. 若平面上的点均匀分布,则 q 可视为常数. 此时,用聚焦算法在域 Ω 中搜索到一个合适的扩展点所花费的时间也将是一个常数,也即时间复杂度为 $O(1)$. 考虑到当生成的三角形越来越多时,平面上也就有越来越多的点,因其扩展状态由 TRUE 变为 FALSE 而退出计算. 综上所述,扩展 AB 边时所花费的时间主要由找初始扩展点 D_0 决定,其时间复杂度为 $S_3 = O(\log_2 m) < O(\log_2 n)$. 式中,$m$ 为对扩展新三角形有贡献的点数. 通常,平面上 n 个点集三角剖分的三角形数可计算出来,且为 $T(n) = 2n - P - 2$,其中 P 为边界点数[8](位于凸壳上的点). 再据图论可知,在这些三角形中共有 $E(n) = T(n) + n - 1 = 3n - P - 3$ 条边. 由于每条边只扩展一次,可推知扩展三角形并连成三角网的时间复杂度为

$$S_4 = E(n)\, S_3 < O(n\log_2 n).$$

由此可知,整个三角剖分的算法复杂度 $S = S_2 + S_4$,最坏情况为 $O(n^2)$,一般情况为 $O(n\log_2 n)$.

4 结束语

本文设计了存贮平面上散乱数据点的数据结构. 查询时,首先用二分法在 X 索引数组中根据要查询点的 x 坐标,找到它所属的类,再用同样的方法在该类所对应的 Y 索引数组中去查找. 它解决了在平面上散乱点集中点的查找问题,且速度快,特别适合于散乱点较多的情况. 在此基础上,根据 Delaunay 三角剖分的"圆准则",提出了算法复杂度为 $O(n\log_2 n)$ 的三角网自动连接的聚焦算法. 给出的算例表明,该算法十分有效,特别适合于大数据量的三角剖分.

参 考 文 献

1 Schumaker L L. Triangulations in CAGD[J]. IEEE Computer Graphics and Applications, 1993,13(1), 47～52

2 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,(1):48～54

3 Tsai V J D. Delaunay triangulations in TIN creation: An overview and a linear time algorithm[J]. Int. J. of GIS, 1993, 7(6): 501～524

4 潘国荣,王穗辉. 微机图形学及其应用[M]. 上海:同济大学出版社,1997. 115～120

5 王 莉,李宗保,吴志明. 计算机图形学及其在工程中的应用[M]. 北京:人民交通出版社,1992. 240～241

6 凌海滨,吴 兵. 改进的自连接 Delaunay 三角网生成算法[J]. 计算机应用,1999,19(12):10～12

7 毛善君,季景贤. 寻找三角形扩展点的一种有效算法[J]. 中国矿业大学学报,1995,24(4):76～79

8 洪家荣,丁明峰. 三角剖分的模拟退火算法[J]. 计算机学报,1994,17(9):682～689

A Focus Algorithms for Auto-Connected Delaunay Triangular Mesh

Wang Huajun

(College of Information Science and Engineering, Huaqiao University, 362021, Quanzhou, China)

Abstract Algorithms for the automatic generation of Delaunay triangular mesh can basically be classified as: divide and conquer, point-by-point insertion, growth of triangular mesh. Based on the third one, the author proposes a focus algorithm for the automatic connection of Delaunay triangular mesh. The proposed algorithm is also based on circle criterion of Delauney triangulation. Moreover, it is proposed by adopting object-oriented technology and by using dot array and dot index array to store the 2D scattered and disordered data. During expanding new triangle, the proposed algorithm let the searching range of dot be controlled within the circumcircle of given triangle so as to accelerate the computation. The proposed algorithm is very efficicent and is suitable for triangulation of large amount of data in particalar as indicated by the given examples of computation.

Keywords delaunay, triangulation, contour, computational geometry