

一种改进的快速定位非 Delaunay 三角形的逐点插入算法

刘晓健^{1,2,3)}, 单广治³⁾, 裘乐森^{1,2,3)}, 张树有^{1,2,3)*}, 陈思诺³⁾, 丁卓琛³⁾, 丛添瑞³⁾

¹⁾ (浙江大学流体动力基础件与机电系统全国重点实验室 杭州 310058)

²⁾ (浙江省高档数控机床技术创新中心 台州 317500)

³⁾ (浙江大学机械工程学院 杭州 310058)

(zsy@zju.edu.cn)

摘要: 逐点插入算法是计算机图形学复杂表面构建时常用的 Delaunay 三角化方法, 但节点插入时需要遍历全局三角网格, 存在时间效率较低的问题。为了优化时间复杂度, 提出快速定位非 Delaunay 三角形的逐点插入算法。根据 Delaunay 三角形的基础性质, 提出并证明了非 Delaunay 三角形依次相邻定理和最近点为嫌疑点定理, 并基于定理快速定位非 Delaunay 三角形; 然后分析并解决节点规模庞大且分布密度不均匀时的局部坏死现象, 提出改进的不需要更新遍历阈值的逐点插入算法, 通过对不同分布特征和数量规模的二维点集进行测试, 其时间复杂度在理想情况下可以达到 $O(N \log N)$ 。对于随机生成的点集, 相较于传统优化算法, 所提算法三角化 100 万点的时间效率提升达到了 46.96%, 且用时与点数更接近线性关系。实验结果表明, 与 Bowyer-Watson 算法、传统优化算法相比, 所提算法在三角化效果完全一致的情况下显著提高了时间效率, 且对不同规模和分布特征的数据集均具有良好的适应性和鲁棒性。

关键词: Delaunay 三角化; 逐点插入法; 时间复杂度; 算法改进; 快速定位非 Delaunay 三角形
中图法分类号: TP391.41 **DOI:** 10.3724/SP.J.1089.2024-00176

An Improved Algorithm for the Rapid Localization of Non-Delaunay Triangles through Point-by-Point Insertion

Liu Xiaojian^{1,2,3)}, Shan Guangzhi³⁾, Qiu Lemiao^{1,2,3)}, Zhang Shuyou^{1,2,3)*}, Chen Sinuo³⁾, Ding Zhuochen³⁾, and Cong Tianrui³⁾

¹⁾ (State Key Laboratory of Fluid Power & Mechatronic Systems, Zhejiang University, Hangzhou 310058)

²⁾ (Zhejiang Advanced CNC Machine Tool Technology Innovation Center, Taizhou 317500)

³⁾ (School of Mechanical Engineering, Zhejiang University, Hangzhou 310058)

Abstract: The incremental insertion algorithm is a commonly used Delaunay triangulation method for constructing complex surfaces in computer graphics. However, inserting nodes requires traversing the entire global triangular mesh, which leads to low time efficiency. To optimize the time complexity, a fast method for locating non-Delaunay triangles in incremental insertion algorithms is proposed. Based on the fundamental properties of Delaunay triangles, the “Adjacent Non-Delaunay Triangle Theorem” and the “Nearest Point as Suspect Theorem” are introduced and proven, which facilitate the rapid identification of non-Delaunay triangles. Furthermore, the issue of local necrosis occurring in scenarios with large node scales and uneven distribution density is analyzed and resolved, introducing an improved incremental inser-

收稿日期: 2024-04-10; 修回日期: 2024-08-15。基金项目: 浙江省科技计划 (2023C01166); 国家自然科学基金 (52375271); 浙江省自然科学基金重点项目 (LZ22E050008)。刘晓健 (1983—), 男, 博士, 教授, 硕士生导师, 主要研究方向为产品数字化设计、复杂曲面加工; 单广治 (1999—), 男, 硕士研究生, 主要研究方向为数控机床设计; 裘乐森 (1977—), 男, 博士, 教授, 博士生导师, 主要研究方向为产品定制设计、智能制造; 张树有 (1963—), 男, 博士, 教授, 博士生导师, 论文通信作者, 主要研究方向为数字化设计与制造、计算机辅助几何设计; 陈思诺 (2001—), 男, 硕士研究生, 主要研究方向为计算机视觉; 丁卓琛 (2000—), 男, 硕士研究生, 主要研究方向为复合制造; 丛添瑞 (2001—), 男, 硕士研究生, 主要研究方向为路径规划。

tion algorithm that does not require updating the traversal threshold. Tests conducted on two-dimensional point sets with various distribution characteristics and sizes show that the time complexity can ideally reach $O(N \ln N)$. For randomly generated point sets, compared to traditional optimized algorithms, the proposed algorithm achieves a 46.96% improvement in time efficiency for triangulating one million points, and the time usage is closer to a linear relationship with the number of points. Experimental results indicate that the proposed algorithm significantly enhances time efficiency while maintaining identical triangulation outcomes compared to the Bowyer-Watson algorithm and traditional optimized algorithms. It also demonstrates good adaptability and robustness across datasets of varying scales and distribution characteristics.

Key words: Delaunay triangulation; point-by-point insertion algorithm; time complexity; algorithm improvement; rapidly identifying non-Delaunay triangles

1 相关工作

计算机图形学中常用网格来模拟复杂物体的表面、多边形网格和三角网格,其中,多边形网格存在结构复杂且顶点不共面的问题,而三角网格稳定性强、结构简单,可以快速生成,所以在处理真实世界的图形时往往需要将图形转化为三角网格^[1-2]。Delaunay三角化是生成三角网格的重要方法,Delaunay三角网格是Voronoi图的伴生图形,Delaunay三角形具有2个性质:(1)最大空圆性质:任意一个Delaunay三角形的外接圆不包括其他顶点;(2)最小角最大性质:对于同一个点集所能生成的所有三角网格,Delaunay三角网格中三角形的最小角是最大的。Delaunay三角化的优势在于尽可能地避免出现狭长三角形,使得三角网的构建更加合理与准确,Miles^[3]与Lingas^[4]已经证明:在一般情况下,Delaunay三角网格是最优的,Delaunay三角网格在处理复杂地形以及点云分割等领域应用广泛。

目前,有3种主流的Delaunay三角化方法^[5]:分治算法、三角网格生长法和逐点插入法。分治算法与三角网格生成法属于静态三角网格生成算法,逐点插入法属于动态三角网格生成算法。大部分Delaunay三角化方法采用逐点插入法。20世纪70年代,Lawson^[6-7]率先提出逐点插入的基本实现方法(Lawson算法),如图1所示;随后,许多研究者对该算法进行了优化,Bowyer-Watson算法^[8-9]是较为重要的优化算法,该算法减少了搜索坏边的流程,在较长一段时间内成为逐点插入法中最常用的算法,其流程如图2所示。

逐点插入法实现简单、占用内存少,但由于需要随着节点的不断插入动态地更新三角网格,因此

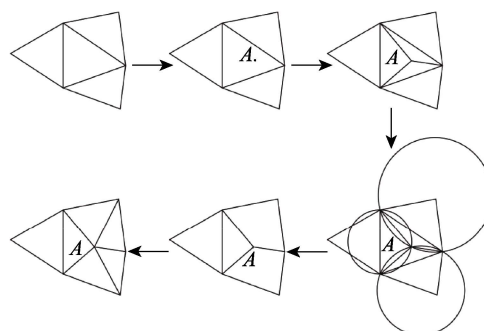


图1 Lawson算法^[6-7]流程示意

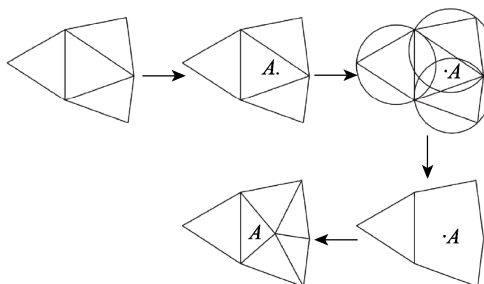


图2 Bowyer-Watson算法^[8-9]流程示意

存在时间效率低的缺陷。其中, Lawson算法中每插入一个节点,需要遍历已存在的三角形确定节点所处的三角形; Bowyer-Watson算法中每插入一个节点,需要遍历所有三角形外接圆判断该点是否位于这些外接圆内:随着数据集不断扩大,三角化速度将越来越慢,因而难以处理庞大的数据集。过去数十年间,为了提高逐点插入法的时间效率,研究者提出了各种优化方法。文献[10]中,定义插入节点与任意三角形的3条边构成的矢量面积为 S_1 、 S_2 和 S_3 ,面积坐标为 L_1 、 L_2 和 L_3 ,顶点按逆时针排列;若 L_1 、 L_2 和 L_3 均大于0,则点位于三角形内部;若任意面积坐标为0,则点在三角形边或顶点上;若存在面积坐标小于0,则点位于三角形外部;在

小于 0 的面积坐标对应边的方向搜索包括插入节点的三角形,以减少插入每个节点时需要搜索的三角形数量。该算法的缺点是当出现 2 条边对应的面积坐标为 0 时,存在 2 条搜索路径,只能选择一个方向来搜索。文献[11]对数据点集进行分格管理,提出一种基于线线关系的最速方向定位法,提高了搜索插入节点所在三角形的速度。文献[12]提出对数据点进行预排序,并基于三角形重心和插入点相对位置搜索插入节点所在三角形的方法,进一步缩短了搜索路径。文献[13]将分治算法与逐点插入法相结合,用分割阈值划分点集,分别对每个子集三角化,最后合并所有子集,平衡了 2 种方法的优缺点,提高了传统逐点插入法算法的时间效率;但在实际操作时不易选取合适的分割阈值,若取值较大,则占用内存提高,子网间的合并变得复杂,否则,时间效率提升不明显。文献[14]采用分治-逐点插入算法,将通过面积坐标定位搜索点所在三角形方向的思想引入子集的逐点插入部分,大大提高了每个子集的三角化速度。文献[15]将 Bowyer-Watson 算法中形成凸包后搜索相邻三角形的复杂过程替代为排序过程,解决程序非正常中断的问题,简化数据结构,提高了计算效率。文献[16]提出基于弦内角的计算方法,简化判断点是否在三角形外接圆内的计算步骤,提高了时间效率。针对将分治思想引入逐点插入法增加了编程的难度的问题,文献[17]对扩展三角形、凹边界填充和子网合并这 3 个步骤进行优化,并设计了算子,简化了算法在程序中的实现。文献[18]将分治构网的思想引入并行计算,基于多核处理器实现对庞大数据集的快速三角化,但仍存在占用内存较大、依赖处理器性能的缺点。文献[19]在分治构网和并行计算的基础上,引入逐点插入算法对每个子集进行三角化,提高了传统并行算法的时间效率和加速比。

针对逐点插入法时间效率较低的问题,当前研究主要有 2 种优化思路:一是引入分治算法的思想,将庞大的数据集划分为较小的子集,即分治-逐点插入算法;二是优化插入节点时的搜索流程,快速找到插入节点所在的三角形,通过局部优化程序(local optimization procedure, LOP)进行局部优化,即缩短路径优化算法。但是,分治-逐点插入算法存在分割阈值不易选取的问题,且对每个子集三角化的速度是限制算法时间效率的最主要因素,通过优化逐点插入算法的时间复杂度,可以显著地提高分治-逐点插入算法的时间效率;而缩短

路径优化算法中,随着数据集的不断扩大,插入每个节点时需要搜索的三角形仍然会逐渐增多,时间复杂度相应升高。因此,提高逐点插入法本身的时间效率仍具有意义。

本文根据 Delaunay 三角形的性质提出并证明相邻定理和嫌疑点定理,基于以上 2 个定理提出一种快速定位非 Delaunay 三角形的方法,以降低逐点插入法本身的时间复杂度;通过避免全局遍历以及在插入点局部优化三角网格,降低算法的时间复杂度。

2 相邻定理和嫌疑点定理

本节提出定理,证明给出在三角网格局部应用 Bowyer-Watson 算法的理论基础。定义基本概念如下。

点接三角形: 2 个三角形仅共用一个顶点,称 2 个三角形点接,与三角形 I 点接的三角形 II 称为三角形 I 的点接三角形,如图 3a 所示。

外侧边: 三角形 I 与三角形 II 点接,三角形 II 不包含三角形 I 顶点的边 DE 为三角形 II 相对于三角形 I 的外侧边,如图 3a 所示。

边接三角形: 2 个三角形共用一条边,称 2 个三角形边接,与三角形 I 边接的三角形 II 称为三角形 I 的边接三角形,如图 3b 所示。

非 Delaunay 三角形: 外接圆包含插入节点的三角形。

嫌疑点: 属于非 Delaunay 三角形顶点的点。

外侧弧段: 圆与封闭图形相交且仅有 2 个交点,圆在封闭图形外部的圆弧,如图 4 所示。

内侧弧段: 圆与封闭图形相交且仅有 2 个交点,圆在封闭图形内部的圆弧,如图 4 所示。

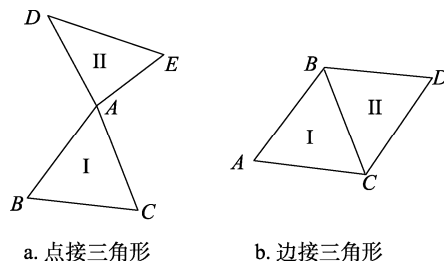


图 3 相邻三角形位置关系示意

2.1 非 Delaunay 三角形依次相邻定理

定理 1: 对于 Delaunay 三角网格,每插入 1 个节点,所有外接圆包含该节点的三角形必然依次相邻。

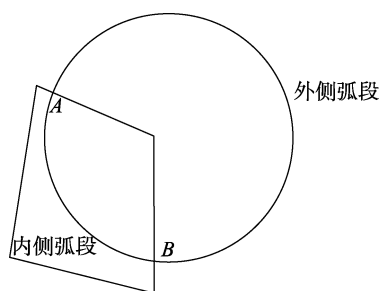
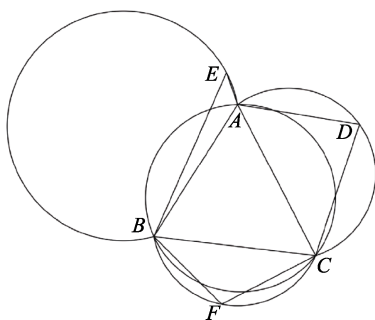


图4 外侧弧段和内侧弧段示意

证明:

(1) 从 Delaunay 三角网格中取 1 个非边界的 $\triangle ABC$ 及其所有边接三角形与点接三角形, 顶点 A 、 B 、 C 与边 AB 、 BC 、 AC 不会与三角网格以外的区域接触。 $\triangle ABC$ 的边接三角形的另 1 个顶点 (D 、 E 、 F) 必然落在 $\triangle ABC$ 外接圆外侧。对于 $\triangle ACD$, 外接圆 ACD 经过 A 、 C 和 D 这 3 个点, 由几何关系知, AC 弧段包含在 ADC 弧段内侧; 同理可知, AB 弧段和 BC 弧段分别位于对应边接三角形弧段 AEB 和 BFC 内侧, 则 $\triangle ABC$ 外接圆完全包含在 3 个边接三角形外接圆外侧弧段相连构成的区域之内。因此, 在 $\triangle ABC$ 外侧插入新节点后, 不存在 $\triangle ABC$ 外接圆包含该点而相邻边接三角形外接圆不包含该点的情况, 如图 5 所示。

图5 $\triangle ABC$ 外接圆完全包含在边接三角形外接圆外侧弧段相连构成的区域之内

对于边界三角形, 由于逐点插入法在开始插点之前需要生成 1 个能容纳所有待插入节点的超级多边形, 因此后续节点必定在超级多边形内部插入, 插入节点位于三角网格内部。如图 6 所示, $\triangle ABC$ 为边界三角形, 插入节点 P 必定位于 $\triangle ABC$ 边接三角形一侧 (直线 AC 右侧区域), 此时不会出现 $\triangle ABC$ 外接圆包含点 P 而边接三角形外接圆不包含该点的情况。

(2) $\triangle ABC$ 所有点接三角形的外侧边顺次相连, 构成 $\triangle ABC$ 的包围圈。根据空圆性质, 除点 A 、 B 、 C 外, 所有 $\triangle ABC$ 的点接三角形的顶点必然

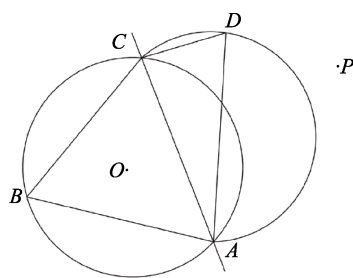
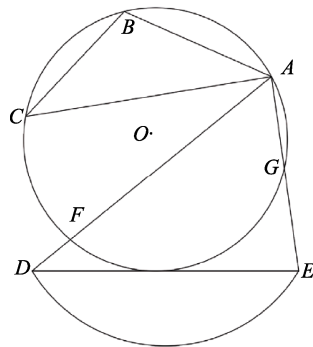
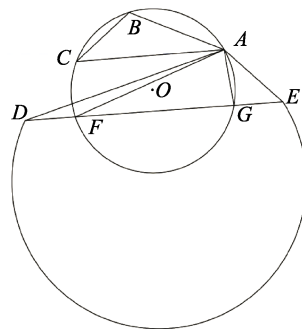


图6 新插入节点位于边界三角形边接三角形一侧

落在 $\triangle ABC$ 外接圆外部, 相邻 2 个顶点连接形成点接三角形外侧边 DE 。此边与 $\triangle ABC$ 的外接圆存在相离、相切和相交 3 种位置关系, 对于相切和相离的情况, 线段 DE 与 $\triangle ABC$ 外接圆不存在交点或仅存在 1 个交点, 点接三角形的外接圆被外侧边隔开的弧段 DE 必然位于 $\triangle ABC$ 外接圆外侧, 如图 7 所示。

图7 线段 DE 与 $\triangle ABC$ 外接圆相切

对于相交的情况, 由于点 D 和 E 均位于 $\triangle ABC$ 外接圆外侧, 因此线段 DE 与 $\triangle ABC$ 外接圆必定存在 2 个交点。如图 8 所示, $\triangle ABC$ 取 DE 边与圆的交点 F 和 G , 观察 $\triangle ADE$ 和 $\triangle AFG$, $\angle DAE$ 大于 $\angle FAG$, 故弧段 FG 必定位于弧段 DE 内侧。综上所述, $\triangle ABC$ 外接圆完全包含在其点接三角形外接圆外侧弧段中。在 $\triangle ABC$ 外侧插入新节点后, 不存在 $\triangle ABC$ 外接圆包含该点而点接三角形外接圆不包含该点的情况。证毕。

图8 线段 DE 与 $\triangle ABC$ 外接圆相交

2.2 最近点为嫌疑点定理

定理 2: 对于 Delaunay 三角网格, 新插入节点 A 距离最近的点 B 为嫌疑点, 即在以点 B 作为顶点的所有三角形中, 必然有至少 1 个三角形为非 Delaunay 三角形。

证明:

对于插入节点 A , 点 B 为距离点 A 的最近节点, 在以点 A 为圆心、 AB 为半径的区域内不会有其他节点。

在所有包含点 B 的三角形中, 至少有 1 个三角形, 其点 B 的对边 CD 与线段 AB 所在的直线是相交的, 点 A 位于射线 BC 和 BD 之间 (若恰好位于线段 BC 或 BD 上, 则 $\triangle BCD$ 的外接圆包含点 A)。此时, 线段 CD 与圆 A 存在相交、相切和相离 3 种位置关系。首先讨论相离和相切的情况, 如图 9 所示, 点 A 必然包含在 $\triangle BCD$ 内部, 即 $\triangle BCD$ 外接圆包含点 A , 为非 Delaunay 三角形。

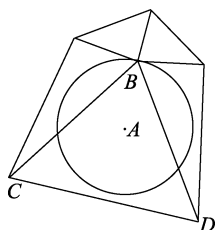


图 9 相离情况

图 10 所示为相交的情形, 点 B 位于圆上, 点 C 和 D 分居以点 A 为圆心, AB 为半径的圆两侧, 且位于外侧, $\triangle BCD$ 的外接圆一定包含圆 A , 则必然包含点 A , $\triangle BCD$ 为非 Delaunay 三角形。综上所述, 以点 B 作为顶点的所有三角形中, 必然存在至少 1 个非 Delaunay 三角形, 因此点 B 为嫌疑点。

证毕。

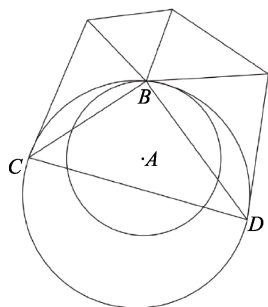


图 10 相交情况

传统的 Bowyer-Watson 算法时间复杂度较高, 每插入 1 个节点, 需要遍历所有三角形的外接圆是否包含该点, 随着三角形数量不断增加, 时间效率

将越来越低。定理 1 说明, 在 Delaunay 三角网格中, 插入节点后出现的非 Delaunay 三角形一定是依次相邻的, 通过找到第 1 个非 Delaunay 三角形并不断地遍历非 Delaunay 三角形的边接三角形或者点接三角形, 就可以找到所有非 Delaunay 三角形。对于每个插入节点, 绝大部分三角形是冗余的, 其外接圆不会包含该点; 而对于每个三角形, 其外接圆大小是固定的。因此已有三角形距离插入节点越近, 其为非 Delaunay 三角形的概率越大。

3 本文算法

3.1 遍历阈值更新的逐点插入算法

在 Delaunay 三角网格中, 由于绝大部分三角形的外接圆不会包含新插入节点, 且非 Delaunay 三角形常位于插入节点附近, 因此可以设定范围, 对于每个新插入节点, 只需要遍历设定范围的三角形以找到非 Delaunay 三角形。对于需要设定的范围, 应根据当前网格的三角网络情况进行选取。遍历阈值更新的逐点插入算法的核心思想为: 首先设定初始遍历阈值 λ , 插入节点后寻找距离该点最近的节点簇, 以遍历阈值 λ 的值为数量上限, 找到这些节点所在的三角形并判断其是否为非 Delaunay 三角形; 不再遍历距离较远的三角形, 根据初步三角化的效果提高阈值或降低阈值, 以提高时间效率。假设需要三角化的数据集 N 包含 n 个节点坐标, 设定初始遍历阈值 $\lambda = M$, 队列 W 用于存储非 Delaunay 三角形, 队列 T 用于存储已插入的节点。基于遍历阈值优化的逐点插入算法流程如图 11 所示。

分别采用遍历阈值更新的逐点插入算法和 Bowyer-Watson 算法, 三角化不同规模点集的时间对比如表 1 所示。可以看出: (1) 随着数据集规模的不断扩大, 包含初始多边形顶点的三角形不断增多。由于这些三角形不易被遍历到, 因此遍历阈值需要随数据集规模的增大而增大, 遍历阈值越大, 算法鲁棒性越强, 但时间复杂度增加。(2) 当数据集规模较小时, 采取较小遍历阈值可以提高时间效率, 与 Bowyer-Watson 算法相比, 遍历阈值更新的逐点插入算法的时间效率并无明显提高; 随着数据集规模的增大, 2 种算法的运行时间差距越来越明显。

3.2 非 Delaunay 三角形引起的局部坏死现象分析

经过对花朵造型点集、树木造型点集和随机分

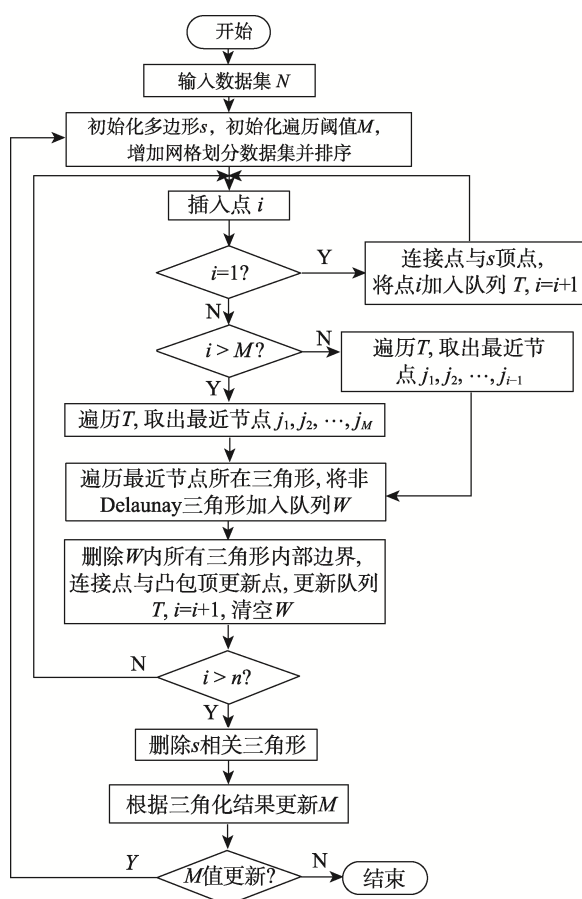


图 11 遍历阈值更新的逐点插入算法流程

表 1 2 种算法三角化不同规模点集的时间对比 s

点集规模	运行时间	
	Bowyer-Watson 算法 ^[8-9]	遍历阈值更新的逐点插入算法
1000	0.912	0.983
2000	3.831	1.810
5000	22.885	5.188
10000	91.352	12.337

布数据点集等具有不同分布特征, 以及不同数据规模的二维数据点集进行三角网格化发现, 遍历阈值更新的逐点插入算法适用于密度分布均匀的数据集, 但在处理数据点分布密度不均匀的数据集时会出现局部坏死现象。如图 12 所示, 圈定区域内的三角网格出现交叉现象, 将超过 2 个三角网格发生的三角网格交叉现象定义为局部坏死现象。

在实际应用中, 需要根据数据集的结构选择合适的遍历阈值。对于较大规模的数据集, 应适当增大遍历阈值, 否则会出现局部坏死现象。这是因为如果插入节点附近的其他节点的局部密度不均匀且遍历阈值较小, 则遍历点集中在一侧, 此时会遗漏本该删除的三角形, 更新剖分结果后这些三角形

不再满足 Delaunay 三角形的性质, 如图 13 和图 14 所示, 后续在这些地方插入节点时, 可能会出现三角网格交叉现象。多次仿真结果显示, 局部坏死往往发生在三角网格边界处或是数据点分布不均匀处。

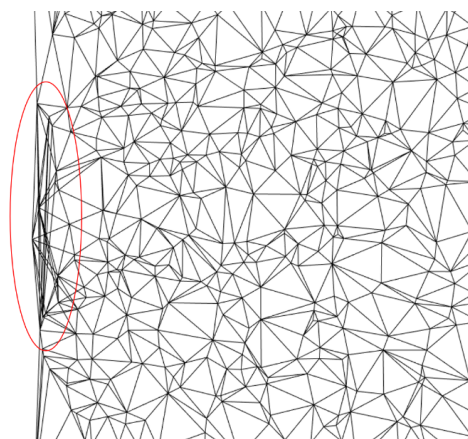
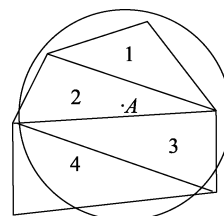
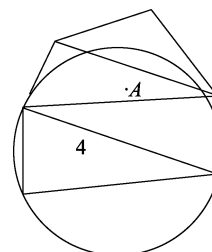


图 12 非 Delaunay 三角形引起的局部坏死现象

图 13 节点 A 位于圆内时 $\triangle 4$ 被遗漏图 14 节点 A 位于 $\triangle 4$ 外接圆内部

由于局部坏死现象引起的初始遍历阈值不易设定的问题降低了算法的实用性, 需要结合最近点为嫌疑点定理进一步优化算法, 以提高算法对不同规模和分布特征的数据集的适应性。

3.3 结合嫌疑点定理的逐点插入算法改进

结合嫌疑点定理进一步优化算法, 插入节点后, 通过最近邻点快速定位到第 1 个非 Delaunay 三角形, 并依次遍历非 Delaunay 三角形的边接三角形, 不断重复遍历的过程, 直到非 Delaunay 三角形的边接三角形中不再出现新的非 Delaunay 三角形。与传统的优化算法相比, 本文算法的优点是省去了定位插入节点所在三角形这一步骤。结合嫌

疑点定理的非 Delaunay 三角形快速定位算法的流程如图 15 所示。

优化后的算法是遍历阈值设定为 1 时的特殊情况, 不需要根据点集的分布特征调整遍历阈值, 可以适应不同分布特征的点集; 该算法大大减少了需要遍历的三角形数量, 每次插入节点需要遍历的三角形数目稳定, 显著地提高了时间效率, 同时可以精准地定位所有非 Delaunay 三角形, 不会出现非 Delaunay 三角形干扰后续插点的情况, 解决了局部坏死问题。

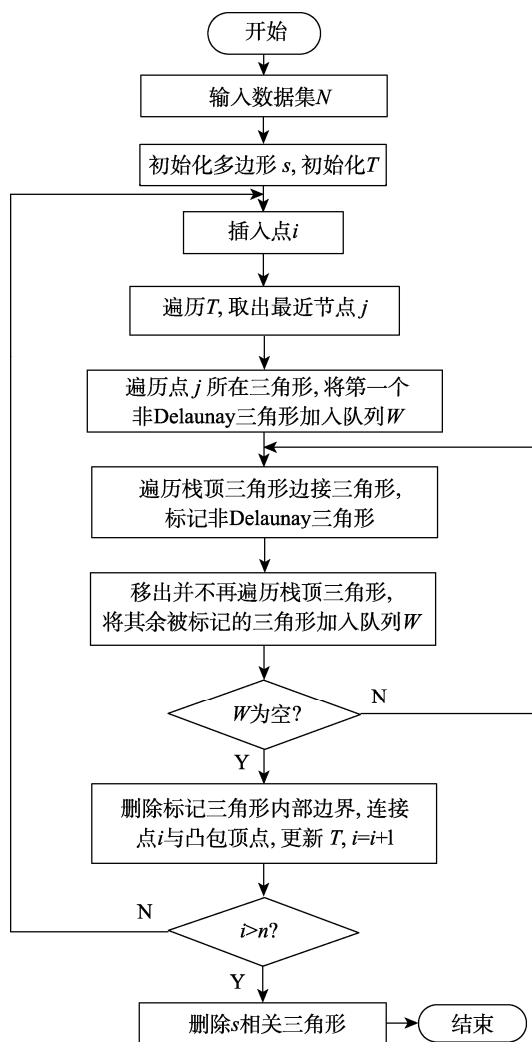


图 15 结合嫌疑点定理的快速定位非 Delaunay 三角形算法流程

改进算法中, 影响时间复杂度的步骤如下: 对数据点分块并重新排序; 每插入一个节点, 搜索所有需要删除的非 Delaunay 三角形; 连接插入节点与删除非 Delaunay 三角形形成的凸包顶点生成新三角形。重新排序数据集的时间复杂度为 $O(NlbN)$, 每插入一个节点, 遍历到的三角形和重新生成的三

角形数目基本是稳定的, 可分别设为 a 和 b , 则这 2 部分的时间复杂度为 $O(aN)$ 和 $O(bN)$, 故总的时间复杂度为 $O(NlbN) + O(aN) + O(bN)$ 。由于计算时间复杂度时只需要考虑最大项, 因此本文算法的时间复杂度为 $O(NlbN)$ 。

4 实例与结果分析

采用 Matlab 实现本文算法。实验环境如下: AMD Ryzen 7 6800H CPU, 16GB 内存。文献[12]算法是目前较为先进的缩短路径优化算法, 该算法对数据点进行预排序, 每次插入节点后仅需要遍历上次循环新生成的三角形, 进一步缩短了插入节点后搜索节点所在三角形的路径, 时间效率较高, 可以作为本文算法的对比算法。通过对不同分布特征的点集进行三角化并记录 CPU 运行时间, 比较本文算法、Bowyer-Watson 算法^[8-9]和文献[12]算法的时间效率。

图 16 所示为 Delaunay 三角网格中插入节点后不同性质的三角形, 其中, $\triangle 1 \sim \triangle 10$ 表示插入节点 P 后本文算法实际遍历到的三角形, $\triangle 1 \sim \triangle 4$ 表示需要删除的非 Delaunay 三角形。可以看出, 传统的逐点插入法需要遍历所有已存在的三角形才能确定需要删除的三角形, 而本文算法仅需从 $\triangle 1 \sim \triangle 10$ 中选择需要删除的三角形; 与本文算法相比, 文献[12]算法增加了寻找插入节点 P 所在三角形这一步骤, 时间复杂度会随搜索三角形数量的增加而增加。表 2 所示为本文算法和 Bowyer-Watson 算法三角化不同点集的效果对比, 其中, 点集 1 包含 2 000 个随机点, 点集 2 为包含 1 000 个点的花朵造型点集, 点集 3 为包含 1 000 个点的树木造型点集。图 17 所示为本文算法三角化 10 万个点的效果。

表 3 所示为采用本文算法、Bowyer-Watson 算法和文献[12]算法时, 三角化不同点集的时间对比。可以看出: (1) 随机点从 1 000 点~10 万点的

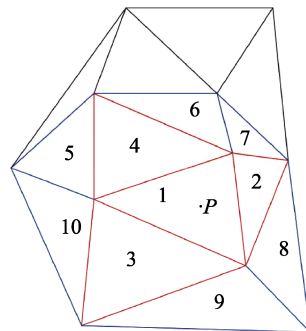
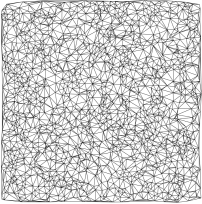
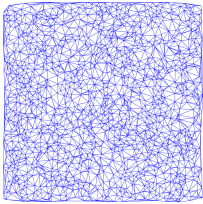
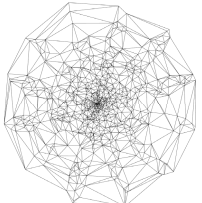
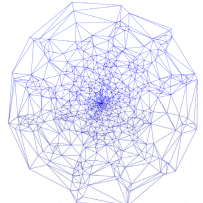
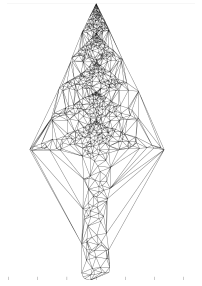
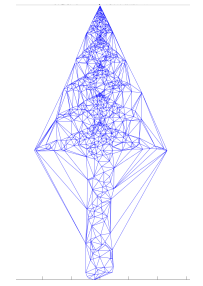


图 16 插入新节点后具有不同性质的三角形

表 2 2 种算法对不同点集三角化效果对比

点集		Bowyer-Watson ^[8-9]	本文
类型	数量		
1	2000		
2	1000		
3	1000		

过程中, Bowyer-Watson 算法^[8-9]的运行时间显著增加, 时间复杂度接近 $O(N^2)$; 本文算法的运行时间基本上随着数据规模的增长呈线性增加, 比 Bowyer-Watson 算法的时间效率提升超过 90%。

(2) 对于 10 000 点以内的数据规模, 与文献[12]算法相比, 本文算法的时间提升效率稳定在 40% 左右, 随着数据规模的增长, 时间提升效率增大到 46.96%, 这是由于随着三角网格数量不断增加, 搜索插入节点所在三角形的路径不断延长, 所以文献

[12]算法的时间效率逐渐降低, 而本文算法省略了这一步骤, 因此时间效率有所提高; 由于本文算法时间复杂度的提升与数据点集数量的增加更加接近线性关系, 因此更适用于处理大规模数据点集。

(3) 同样是 1 000 个点的数据规模, 对于三角化花朵造型点集和树木造型点集, 本文算法的时间效率有所降低。这是因为分布不均匀的数据点生成的 Delaunay 三角网格中的三角形较为狭长, 每次插入节点后需要删除的非 Delaunay 三角形数量有所增加, 需要检查的三角形数目也相应增加, 故时间效率会降低。理论上, 本文算法的最坏情况是, 每次插入新节点都需要删除所有已经存在的三角形重新构造三角网格, 则本文算法退化为 Bowyer-Watson 算法, 算法的时间复杂度增加到 $O(N^2)$, 但在实际应用中通常不会出现这种极端情况。综上所述, 本文算法在处理分布密度均匀的数据点集时, 时间效率较高; 处理分布密度不均匀的数据点集时, 时间效率有所下降。

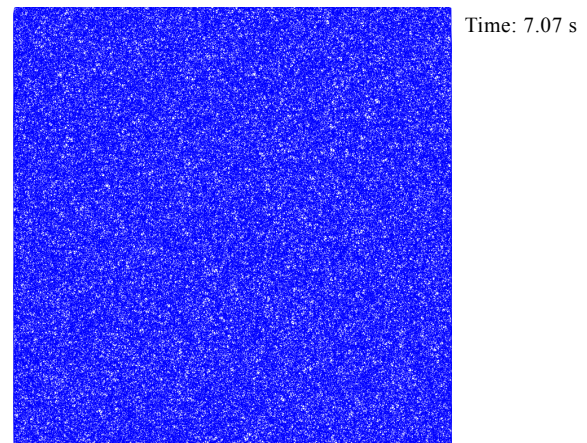


图 17 10 万个点三角化效果展示

表 3 3 种算法三角化不同点集的时间对比

点集		三角形数量	运行时间/s			相比文献[12]算法 时间效率提升/%
特征	数量		Bowyer-Watson ^[8-9]	文献[12]	本文	
随机	1 000	1 959	0.912	0.109	0.066	39.45
随机	2 000	3 952	3.831	0.223	0.135	39.46
随机	5 000	9 939	22.885	0.560	0.336	40.00
随机	10 000	19 923	91.352	1.139	0.686	39.77
随机	100 000	199 833	9 122.251	12.531	7.072	43.56
随机	1 000 000	1 999 664	908 513.664	135.842	72.057	46.96
花朵造型 1 000 个	1 000	1 979	0.956	0.114	0.072	36.84
树木造型 1 000 个	1 000	1 969	0.948	0.117	0.073	37.61

5 结 语

针对传统逐点插入算法时间效率较低的问题, 本文提出一种改进的快速定位非 Delaunay 三角形的逐点插入算法。

(1) 提出并证明了非 Delaunay 三角形依次相邻定理。在 Delaunay 三角化过程中插入节点时需要删除的三角形具有相邻传递性, 且这种传递性会随着距离增加而不断减弱, 因此, 逐点插入算法插入节点的过程中不必遍历目前已经存在的所有三角形, 这是对逐点插入法局部优化的理论依据。

(2) 提出并证明了最近点为嫌疑点定理。以距离插入节点最近的点为顶点构造的所有三角形中, 必然有至少 1 个三角形其外接圆包含该插入节点, 此定理表明, 定位插入节点所在三角形这一步骤实际上是不必要的, 优化方法不必局限于寻找节点所在三角形的最短路径。

(3) 基于相邻定理和嫌疑点定理提出快速定位非 Delaunay 三角形的逐点插入算法。首先结合相邻定理提出遍历阈值更新的逐点插入算法, 分析优化过程中的局部坏死现象并给出解决方案; 进一步, 结合嫌疑点定理, 提出无需更新遍历阈值的快速定位非 Delaunay 三角形的逐点插入算法。通过实例, 验证了本文算法的有效性和鲁棒性; 随着数据集规模不断扩大, 该算法的时间复杂度基本呈现线性上升趋势, 与文献[12]算法相比, 在处理百万点规模的数据点集时, 其在时间效率上提升 46.96%。

传统的逐点插入算法的优化方法通常局限于如何快速定位插入节点所在三角形, 实际上该过程并非必要, 随着三角网格规模的扩大, 搜索路径必然会逐渐延长, 时间效率因此降低。本文提出的通过嫌疑点快速定位非 Delaunay 三角形的算法从底层逻辑上降低逐点插入算法的时间复杂度, 拓宽了提高逐点插入法时间效率的思路。

参考文献 (References):

- [1] Zhang Haojie, Liu Xing, Li Hongjing. Deep Q network-based optimization algorithm for planar Delaunay mesh[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2022, 34(12): 1943-1950(in Chinese)
(张浩杰, 刘星, 李鸿晶. 基于深度 Q 网络的平面域 Delaunay 网格优化算法[J]. *计算机辅助设计与图形学学报*, 2022, 34(12): 1943-1950)
- [2] Zhang Xiaomeng, Lu Zhonghua, Zhang Jian. 3D parallel constrained delaunay mesh generation algorithm and its implementation[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2016, 28(6): 932-942(in Chinese)
(张晓蒙, 陆忠华, 张鉴. 三维并行约束 Delaunay 网格生成算法及实现[J]. *计算机辅助设计与图形学学报*, 2016, 28(6): 932-942)
- [3] Miles R E. Probability distribution of a network of triangles (Mary Beth Stearns)[J]. *SIAM Review*, 1969, 11(3): 399-402
- [4] Lingas A. The greedy and Delaunay triangulations are not bad in the average case[J]. *Information Processing Letters*, 1986, 22(1): 25-31
- [5] Tsai V J D. Delaunay triangulations in TIN creation: An overview and a linear-time algorithm[J]. *International Journal of Geographical Information Systems*, 1993, 7(6): 501-524
- [6] Lawson C L. Generation of a triangular grid with application to contour plotting[R]. Pasadena: California Institute of Technology Jet Pollution, 1972
- [7] Lawson C L. Software for C^1 surface interpolation[M] //Rice J R. *Mathematical Software*. New York: Academic Press, 1977: 161-194
- [8] Bowyer A. Computing Dirichlet tessellations[J]. *The Computer Journal*, 1981, 24(2): 162-166
- [9] Watson D F. Computing the n -dimensional Delaunay tessellation with application to Voronoi polytopes[J]. *The computer journal*, 1981, 24(2): 167-172
- [10] Liu Xuejun, Fu Xinsha, Zhao Jiansan. A study of algorithm for fast constructing triangulation irregular net (TIN)[J]. *China Journal of Highway and Transport*, 2000, 13(2): 31-36(in Chinese)
(刘学军, 符锌砂, 赵建三. 三角网数字地面模型快速构建算法研究[J]. *中国公路学报*, 2000, 13(2): 31-36)
- [11] Liu Shaohua, Wu Dongsheng, Luo Xiaolong, et al. Research on algorithms of point fast position in Delaunay triangular net[J]. *Science of Surveying and Mapping*, 2007, 32(2): 69-70+113(in Chinese)
(刘少华, 吴东胜, 罗小龙, 等. Delaunay 三角网中点目标快速定位算法研究[J]. *测绘科学*, 2007, 32(2): 69-70+113)
- [12] Xu Y B, Liu K X, Ni J Y, et al. 3D reconstruction method based on second-order semiglobal stereo matching and fast point positioning Delaunay triangulation[J]. *PLoS One*, 2022, 17(1): Article No.e0260466
- [13] Wu Xiaobo, Wang Shixin, Xiao Chunsheng. A new study of Delaunay triangulation creation[J]. *Acta Geodaetica et Cartographica Sinica*, 1999, 28(1): 28-35(in Chinese)
(武晓波, 王世新, 肖春生. Delaunay 三角网的生成算法研究[J]. *测绘学报*, 1999, 28(1): 28-35)
- [14] Wu Yuxiao, Zhang Dengrong. Improved algorithm for building Delaunay triangulation[J]. *Journal of Zhejiang University (Science Edition)*, 2004, 31(3): 343-348(in Chinese)
(吴宇晓, 张登荣. 生成 Delaunay 三角网的快速合成算法[J]. *浙江大学学报(理学版)*, 2004, 31(3): 343-348)
- [15] Liu Shihe, Luo Qiushi, Huang Wei. Generation of 2-D non-structural grid by modified Delaunay method[J]. *Engineering Journal of Wuhan University*, 2005, 38(6): 1-5(in Chinese)
(刘士和, 罗秋实, 黄伟. 用改进的 Delaunay 三角化方法生成二维非结构网格[J]. *武汉大学学报(工学版)*, 2005, 38(6): 1-5)
- [16] Xu Hui, Zhang Shuyou. Polygon's triangulation and quick optimum based on angle-of-chord mapping parameters rule[J].

- Journal of Image and Graphics, 2004, 9(4): 445-450(in Chinese)
(徐辉, 张树有. 基于弦内角映射参数的多边形三角剖分快速优化[J]. 中国图象图形学报, 2004, 9(4): 445-450)
- [17] Liu Yonghe, Feng Jinming, Guo Weidong, *et al.* Merging planar Delaunay triangulations based on universal operators and the implementation of a divide-conquer algorithm[J]. Journal of Image and Graphics, 2012, 17(10): 1283-1291(in Chinese)
(刘永和, 冯锦明, 郭维栋, 等. Delaunay 三角网通用合并算子及分治算法的简化[J]. 中国图象图形学报, 2012, 17(10): 1283-1291)
- [18] Li Jian, Li Deren, Shao Zhenfeng. A streaming data Delaunay triangulation algorithm based on parallel computing[J]. Geomatics and Information Science of Wuhan University, 2013, 38(7): 794-798(in Chinese)
(李坚, 李德仁, 邵振峰. 一种并行计算的流数据 Delaunay 构网算法[J]. 武汉大学学报·信息科学版, 2013, 38(7): 794-798)
- [19] Huang Quan, Liu Hao, Liang Pingyuan. Parallel construction algorithm of Delaunay triangulated irregular network[J]. Science of Surveying and Mapping, 2017, 42(6): 171-177(in Chinese)
(黄詮, 刘浩, 梁平元. Delaunay 三角网的并行构网算法[J]. 测绘科学, 2017, 42(6): 171-177)