

基于二维凸壳的平面点集 Delaunay 三角网算法

毕硕本¹ 陈东祺¹ 颜 坚² 郭 忆¹

(南京信息工程大学遥感学院 南京 210044)¹ (南京信息工程大学计算机与软件学院 南京 210044)²

摘 要 提出了一种基于并行二维凸壳算法的平面点集的 Delaunay 三角网生成算法。该算法基于颜坚等在文献[20]中提出的并行二维凸壳算法,在构建凸壳时记录被替换的边和被删除的点,形成一个初始三角网;再在初始三角网的各个三角形内部,采用逐点插入法构建局部的 Delaunay 三角网;最后,对各个局部 Delaunay 三角网的边界边进行局部优化,得到原点集的 Delaunay 三角网。文中给出了算法的正确性说明,实验结果也表明该算法稳定高效。

关键词 凸壳, Delaunay 三角网, 并行算法

中图法分类号 TP301.6 文献标识码 A DOI 10.11896/j.issn.1002-137X.2014.10.066

Planar Delaunay Triangulation Algorithm Based on 2D Convex Hull

BI Shuo-ben¹ CHEN Dong-qi¹ YAN Jian² GUO Yi¹

(School of Remote Sensing, Nanjing University of Information Science & Technology, Nanjing 210044, China)¹

(School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing 210044, China)²

Abstract This paper proposed a planar Delaunay triangulation algorithm based on parallel 2D convex hull algorithm. The proposed algorithm is based on the parallel 2D convex hull algorithm proposed in literature [20] by Jian Yan, etc. It constructs an initial triangulation by recording the grown edges and the removed points during the convex hull building, then constructs partial Delaunay triangulation through point by point method inside each triangle of the initial triangulation, finally locally optimizes the boundary edges of the local Delaunay triangulations to get the Delaunay triangulation of the whole origin set. The correctness of the algorithm was discussed. And the experiment results show that the algorithm is efficient and stable.

Keywords Convex hull, Delaunay triangulation, Parallel algorithm

1 引言

凸壳和 Delaunay 三角网都是计算几何领域的基本几何结构。点集的 Delaunay 三角网的边界就是该点集的凸壳。

Delaunay 三角网生成算法可以分为间接和直接两种。第一类间接算法先生成 Delaunay 三角网的等价 Voronoi 图,再连接图中共边的顶点形成 Delaunay 三角网;第二类间接算法则生成更高维的凸壳,再通过投影转换为 Delaunay 三角网。直接算法没有生成 Voronoi 图或凸壳的中间过程,而是直接生成点集的 Delaunay 三角网。直接算法又可以分为静态生成算法和实时生成算法,静态生成算法又包括生长法、分治法等,实时生成算法允许向现有的 Delaunay 三角网中实时地增加或删除点,逐点插入法是生成 Delaunay 三角网的主要实时算法。

Delaunay 三角网生长法的核心思想是在点集中任取一点,找到与它最近的一点连接成一条线段,然后找到离这条线段最近的点连成三角形,从这个三角形的3条边出发,各自寻找离它们最近的点连成三角形,依次扩展,直到所有的离散点都构造完为止。三角网生长法的各种实现算法多在搜寻第三

点的策略上做文章,比较有代表性的有 Brassel 和 Reif^[1], McCullagh 和 Ross^[2]等。吴佳奇等^[3]提出了一种基于生长法的“弧带搜索排除”方法,通过一系列的圆弧将离散点区域划分成几个弧带,构网时只在当前弧带内搜索第三点。

分治算法思想由 Shamos 和 Hoey 于 1975 年提出^[4]。分治算法最初是被 Lewis 和 Robinson 加以改进并引入到 Delaunay 三角网构网中的^[5]。而后 Lee 和 Schachter 改进和完善了分治算法,在此基础上提出了分割-归并算法^[6],其核心思想是把离散点集划分成两个相互独立的子集,然后每一个子集分别生成各自的 Delaunay 三角网,最后将两者合并为一个完整的 Delaunay 三角网。谢增广^[7]对分治算法的退化情况做了特殊处理。李小丽等^[8]对点集进行格网划分,构建块内子网,再对块进行合并,实现了 Delaunay 三角网的分割归并算法。刘永和等^[9]引入通用算子的概念,简化了分治算法的子网合并过程。

逐点插入算法于 1977 年由 Lawson 提出^[10],随后 Lee、Bowyer 等^[11]众多学者对其进行了改进,其核心思想是先生成一个足以把所有点包围进来的超三角形,随后逐地将点插入到这个大的超三角形中。在插入的过程中,采用 Lawson

到稿日期:2013-12-23 返修日期:2014-03-21 本文受国家自然科学基金项目(41071253,41271410)资助。

毕硕本(1965—),男,博士后,博士生导师,主要研究方向为气象信息融合、气象史料数据挖掘、三维可视化等;陈东祺(1989—),男,硕士生,主要研究方向为气象信息融合、软件开发等,E-mail:cdq19891116@hotmail.com(通信作者)。

提出的局部优化过程(Local Optimization Procedure, LOP)进行优化,以保证最终生成的是Delaunay三角网。李水乡等^[12]采用邻接单元搜索、双向链表存储、随机方向搜索、邻接旋转、几何量继承等技术,实现了一种快速Delaunay逐点插入算法。

近年来,计算平面点集Delaunay三角网的改进算法层出不穷。Loffier和Snoeyink^[13]通过对模糊点集预处理后,在线性时间复杂度内求出了其Delaunay三角网;Lo^[14]提出了三维Delaunay三角网的并行生成算法;Meng Qi等和Rupesh Nasre等^[15,16]分别在图形处理器(Graphics Processing Unit, GPU)上实现了并行的二维Delaunay算法;李根等^[17]对平面上的散乱点进行分块,建立子块索引二叉树,利用Graham扫描技术较快地生成了Delaunay三角网。

鲍蕊娜等^[18]对点集进行预处理,通过凸壳生成构建Delaunay三角网,该算法构网速度更快,并能够在生成网的过程中对共线进行处理;陈学工等^[19]提出了一种基于最优凸壳技术的Delaunay三角网剖分算法,该算法不但避免了所有交点的测试,而且使得新加入点与凸壳边的平均比较次数不大于4,从而实现高效的三角剖分。而本文提出的基于二维凸壳的Delaunay三角网算法则是基于颜坚等^[20]提出的并行二维凸壳算法,通过初始三角网将点集划分为若干个三角形区域进行分治求解,是一种改进的分治建网算法。结果表明,基于二维凸壳的Delaunay三角网生成算法更稳定、高效。

2 算法理论依据

Delaunay三角网实际上是Voronoi图的直线对偶图,其中每个三角形的外接圆不包含点集中的其他任何点。图1描述了Voronoi图与Delaunay三角网的对应关系,其中实线为Delaunay三角网,虚线为Voronoi图。

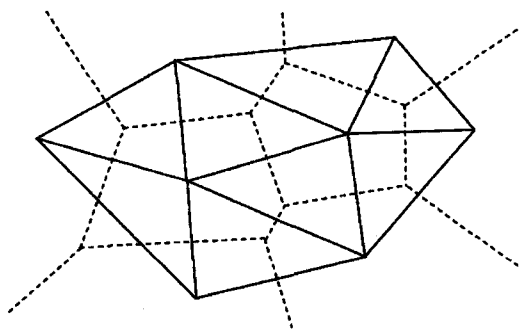


图1 Voronoi图与Delaunay三角网示意图

Delaunay三角网具有如下性质:

- ①唯一性:任意点集的Delaunay三角网是唯一的;
- ②空外接圆性质:任意一个Delaunay三角形的外接圆都不包含点集中的其它任意点;
- ③最大最小角性质:在给定点集的三角划分中,Delaunay三角网的最小三角形内角是最大的。

2.1 局部优化算法

Lawson提出了局部优化过程方法^[21],即由任何剖分准则得到的三角网经过LOP优化,即可确保成为Delaunay三角网。LOP算法的具体步骤如下:

- (1)将具有公共边的两个三角形拼接成一个凸四边形;
- (2)以空外接圆准则检查其中一个三角形,判断四边形的

第四个顶点是否在该三角形的外接圆的内部;

(3)如果(2)判断结果为是,则从Delaunay三角网中删除该条公共边,并加入四边形的另一条对角线。

下面提出2个定理,它们是本文算法的理论支撑。由于下述定理均可通过代数运算得出证明,本文不再赘述其证明过程。

定理1 对于凸四边形ABCD(见图2),如果点D在 $\triangle ABC$ 的外接圆外(内),则点B在 $\triangle ACD$ 的外接圆外(内)。

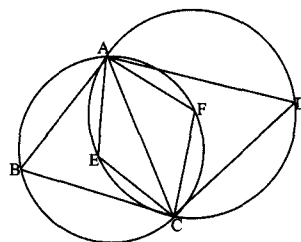


图2 定理1示意图

定理2 连接凸四边形ABCD的对角线(AC或BD)形成三角网,对该三角网进行LOP优化后所得三角网为Delaunay三角网(见图3)。

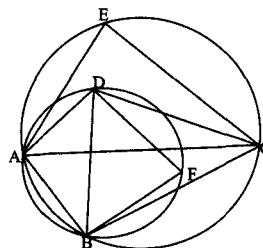


图3 定理2示意图

2.2 Delaunay三角网插入点算法

在插值过程中,需要在已生成的Delaunay三角网的给定三角形内插入新点,生成新的Delaunay三角网。假设待插入内点的三角形为 $\triangle ABC$,插入的新点为P,则插入点算法步骤为:

- (1)分别连接点P和点A、B、C,删除原来的 $\triangle ABC$,加入3个新生成的三角形 $\triangle ABP$ 、 $\triangle PBC$ 、 $\triangle PCA$;
- (2)分别对边AB、BC、CA相对于点P进行迭代LOP优化。

新插入点P后,边AB相对于点P的迭代LOP优化算法步骤为:

- (1)如果AB不是原点集的凸壳边界上的边,则设点Q为与 $\triangle ABP$ 相邻于边AB的三角形的第三个顶点;否则,迭代LOP优化完毕;
- (2)对四边形APBQ进行LOP优化,如果在优化过程中交换了对角线,则迭代地对边AQ、QB相对于点P进行迭代LOP优化;否则,迭代LOP优化完毕。

3 算法描述

3.1 核心理想

首先,对平面点集进行凸壳生长。在生长过程中,不删除被替换的边,这样在边被生长后,原来边与新生成的两条边就形成了一个三角形。当凸壳生长完成时,就形成一个初始三角网。该初始三角网实际上是对原点集凸壳的一个三角剖

分。由于在凸壳生长过程中,已经对所有非凸亮点进行了位置判断,因此容易得到初始三角网中每个三角形内部的所有非凸亮点。然后在各个三角形内部,采用逐点插入法构建局部的 Delaunay 三角网。最后,在整个三角网中,对各个局部 Delaunay 三角网的边界边进行局部优化(LOP),便可得到原点集的 Delaunay 三角网。

3.2 生成初始三角网

初始三角网通过记录点集凸壳生长过程中产生的三角形和被删除的点集数据而得到。

(1)初始凸壳。在寻找点集的初始不完全凸壳点序列时,找到 3 个或 4 个凸亮点,形成一个三角形或四边形作为点集的初始凸壳,并找出了所有在初始凸壳内部的点。当初始凸壳为三角形时,将找到的点作为该三角形的内点集;当初始凸壳为四边形时,连接其一条对角线,将其划分为两个共边的三角形,对找到的点根据其与其与对角线的位置关系分为两个子集,分别作为两个三角形的内点集。

(2)边的生长。在对有向边进行迭代生长时,找到的新的凸亮点与生长边组合为一个新的三角形。对原生长边的外点集合分类后,除两条新生边的外点集外,其余点在新生三角形的内部,将它们作为新生三角形的内点集。

这样,在点集凸壳的求解过程中,临时产生的凸壳边和内部的点不会被删除。最终的凸壳是由若干相邻的三角形拼接而成的三角网,其中每个三角形包含了原点集中所有在其内部而未加入三角网的点。

3.3 局部 Delaunay 三角网生成

局部 Delaunay 三角网的生成即是求初始三角网中各三角形及其内部点集的 Delaunay 三角网。图 4 示出初始三角网中的一个三角形 T 及其所有内点。设 C 为 T 的内点集合,将 T 当作一个只含有一个三角形的 Delaunay 三角网,则局部 Delaunay 三角网生成过程为:从 C 中随机取出一一点 P ,找出 T 中包含 P 的三角形 T_1 ,连接 P 和 T_1 的 3 个顶点并对 T_1 的 3 条边相对于点 P 进行 LOP 优化;重复,直至 C 为空。最终,形成如图 5 所示的局部 Delaunay 三角网。

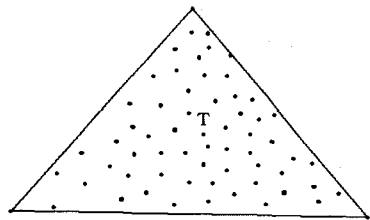


图 4 Delaunay 三角形及其内部点示意图

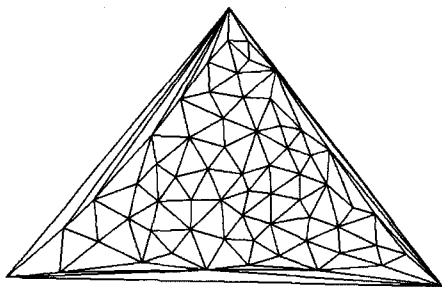


图 5 局部 Delaunay 三角网图

3.4 局部网拼接

局部 Delaunay 三角网生成完成后,初始三角网变成由图 5 所示的若干 Delaunay 三角网拼接成的三角网,对其进行 LOP 优化即可得到原点集的 Delaunay 三角网。局部网的拼接即是要对所有出现在初始三角网中的边进行 LOP 优化,从而完成整个三角网的 LOP 优化。

设 T_1 、 T_2 为两个相邻的局部 Delaunay 三角网, E 为它们的公共边界, A 、 B 为 E 的两个端点,在两个局部 Delaunay 三角网中以 E 为边的三角形的第三个顶点分别为 P_1 、 P_2 ,如图 6 所示。对边 E 进行 LOP 优化的具体步骤为:

(1)将 T_1 、 T_2 拼接为一个三角网;

(2)对边 E 相对于点 P_1 进行 LOP 优化,如果交换了对角线,就找到以 P_1A 、 P_1B 为边的两个新三角形的顶点 Q_1 、 Q_2 ;

(3)对边 P_1A 、 P_1B 分别相对于点 Q_1 、 Q_2 执行第(2)步的操作;

(4)迭代执行(2)、(3)两步,直到 LOP 优化过程不再交换对角线。

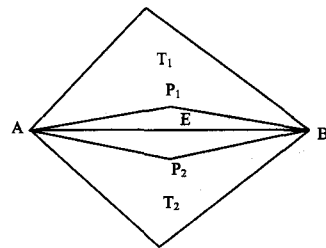


图 6 局部网拼接示意图

4 算法分析

4.1 算法正确性

从 3 个方面说明基于二维凸壳的 Delaunay 三角网算法的正确性:

(1)通过初始三角网和局部三角网生成所得的是原点集的一个三角剖分;

(2)对这个三角网进行 LOP 优化,必然得到其 Delaunay 三角剖分;

(3)局部网中狭长三角形集中在局部网的边界处,因此局部网拼接过程中的 LOP 过程迭代次数少,收敛速度快。

4.2 算法复杂度

根据二维凸壳算法的复杂度分析,本算法在构建初始三角网时的复杂度为 $O(n \log n)$ 。生成局部三角网时,最多向初始三角网中插入 n 个点;每插入一个点,都要在局部 Delaunay 网中对其进行 LOP 优化,由于最多有 $\log n$ 个局部网,因此复杂度为 $O(n \log \log n)$ 。在进行局部 Delaunay 三角网的拼接时,需要对最多 $\log n$ 条边界进行 LOP 优化,时间复杂度为 $O((\log n)^2)$ 。基于二维凸壳的 Delaunay 三角网生成算法的总时间复杂度为 $O(n \log n)$ 。

5 实验分析

为验证算法的效率与分析相吻合,通过实验对其进行了效率分析,并以相同复杂度的分治建网算法作为参考。利用

MATLAB 随机数发生器生成平面上各不重复的固定数量的随机点集作为实验样本数据,各算法的平均执行时间如图 7 所示。

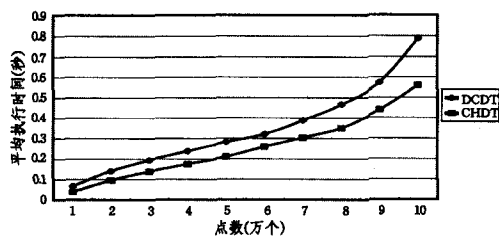


图 7 Delaunay 三角网生成算法执行时间对比图

在图 7 中,DCDT 表示分治算法,它是生成 Delaunay 三角网的算法之一,并被广泛应用,其在文献[22]中被证明具有一定的代表性;CHDT 表示基于二维凸壳的 Delaunay 三角网算法。根据 Delaunay 三角网的唯一性,对于给定的点集,无论采用什么样的算法生成的 Delaunay 三角网都是一样的。由此可见,基于凸壳的 Delaunay 建网算法效率高于分治算法。这一方面说明了本文提出的改进型二维凸壳算法是高效的,尤其是当数据量越来越大时,DCDT 由于需要占用的计算机内存也随之变大,这也导致与 CHDT 差距越来越大;另一方面说明了利用多核处理器进行并行计算的优越性。测试所用 Delaunay 三角网算法的复杂度是相同的,因此图中曲线的变化趋势基本相同。但在曲线的起始处,并未呈现出算法应有复杂度的变化趋势,这有可能是由于实验环境导致的误差。

结束语 本文提出了基于并行二维凸壳算法的 Delaunay 三角网生成算法。在并行求解平面点集的凸壳的过程中,记录被生长的边和被删除的点,生成初始三角网;按初始三角网对未加入网络的点集空间划分,得到若干个包含内点集合的局部三角形;对于各个局部三角形,利用逐点插入法将其所有内点插入到网络中,形成局部 Delaunay 三角网;最终对整个网络进行 LOP 优化,得到原点集的 Delaunay 三角网。算法充分利用并行二维凸壳算法的并行计算优势,减少了计算时间。理论和实验结果表明,本文提出的基于二维凸壳的 Delaunay 三角网算法稳定可靠,且执行效率相对较高,适合于海量点集的求解。本文提出的理论和思想可以应用到三维和更多维的情况,通过实验更好地验证算法本身的理论复杂度也是下一步的研究目标。

参 考 文 献

- [1] Brassel K E, Reif D. A Procedure to Generate TIN Polygons [J]. *Geographical Analysis*, 1979, 11(3): 289-303
- [2] McCullagh M J, Michael J, Ross C G. Delaunay Triangulation of a Random Data Set for its Arithmetic Mapping [J]. *The Cartographic Journal*, 1980, 17(2): 93-99
- [3] 吴佳奇,徐爱功. Delaunay 三角网生长法的一种改进方法[J]. *测绘科学*, 2012, 37(2): 103-187
- [4] Shamos M I, Hoey D. Closest-point Problems [C]//*Proceeding of the 16th Annual IEEE Symposium on Foundation of Computer Science*. Los Angeles, California: IEEE, 1975: 151-162
- [5] Lewis B A, Robinson J S. Triangulation of Planar with Applications [J]. *The computer Journal*, 1978, 21(4): 324-332
- [6] Lee D T, Schacher B J. Two Algorithms for Constructing a Delaunay Triangulation [J]. *International Journal of Computer and Information Sciences*, 1980, 9(3): 219-242
- [7] 谢增广. 平面点集 Delaunay 三角剖分的分治算法[J]. *计算机工程与设计*, 2012, 33(7): 2652-2658
- [8] 李小丽,陈花竹. 基于网格划分的 Delaunay 三角剖分算法研究[J]. *计算机与数字工程*, 2011, 39(7): 57-59
- [9] 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
- [10] Lawson C L. Generation of a Triangular Grid with Application to Contour Plotting [C]//*Proceeding of Technical Memorandum, California institute of technology. Jet Pollution Laboratory*, 1972: 299
- [11] Johnstone J K, Sloan K R. Tensor Product Surfaces Guided by Minimal Surface Area Triangulations [C]//*Proceedings of IEEE Conference on Visualization*. 1995: 354-361
- [12] 李水乡,陈斌,赵亮. 快速 Delaunay 逐点插入网格生成算法[J]. *北京大学学报:自然科学版*, 2007, 43(3): 302-306
- [13] Loffier M, Snoeyink J. Delaunay Triangulation of Imprecise Points in Linear Time after Preprocessing [J]. *Computational Geometry*, 2010, 43(3): 234-242
- [14] Lo S H. Parallel Delaunay Triangulation in Three Dimensions [J]. *Computer Methods in Applied Mechanics and Engineering*, 2012, 237(1): 88-106
- [15] Qi M, Cao T-t, Tan T S. Computing 2D constrained Delaunay triangulation using the GPU [J]. *IEEE Transactions on Visualization and Computer Graphics*, 2013, 19(5): 736-748
- [16] Nasre R, Burtcher M, Pingali K. Morph Algorithms on GPUs [C]//*Proceedings of the 18th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP' 13)*. 2013: 147-156
- [17] 李根,邹志文,鞠时光. 结合二叉树和 Graham 扫描技术的高效 Delaunay 三角网构建算法[J]. *计算机应用研究*, 2010, 27(3): 894-896
- [18] 鲍蕊娜,李向新,麻明,等. 基于凸壳技术的 Delaunay 三角网生成算法研究[J]. *科学技术与工程*, 2011, 11(4): 764-767
- [19] 陈学工,黄晶晶. 基于最优凸壳技术的 Delaunay 三角剖分算法[J]. *计算机工程*, 2007, 33(17): 93-95
- [20] 颜坚,毕硕本,汪大,等. 多核架构下计算凸壳的并行算法[J]. *计算机科学*, 2013, 40(2): 16-19, 57
- [21] Lawson C L. Software for C1 Surface Interpolation [M]//*Mathematical Software III*. New York: Academic Press, 1977
- [22] 武晓波,王世新,肖春生. 一种生成 Delaunay 三角网的合成算法[J]. *遥感学报*, 2000, 4(1): 32-35