文章编号:1671-8860(2010)04-0403-04　　**文献标志码**:A

一种栅格辅助的平面点集最小凸包生成算法

王结臣[1]　陈焱明[1]

(1　南京大学地理信息科学系,南京市汉口路 49 号,210093)

摘　要:针对平面点集的最小凸包生成问题,提出一种栅格辅助的算法,预先剔除那些不可能成为凸包顶点的点,从而提高算法效率,算法的时间复杂度可近似达到 $O(n)$,最坏时间复杂度与 Graham 扫描算法相同。试验表明,随着行列数的增加,计算效率先快速递增,随后逐渐减小;当栅格行列数取值为总点数的平方根时,剔除比接近最大值,算法执行效率亦相对较高。

关键词:最小凸包;栅格化;算法

中图法分类号:P208

平面点集最小凸包是指对于给定的点集合 S,能够包含 S 集合内所有点并且顶点属于 S 集合的最小凸多边形[1]。最小凸包是物体形状描述、特征抽取的一个重要工具,最小凸包算法也被应用于许多领域,如模式识别、数据挖掘、资源配置和图像处理等[1-10]。

为提高算法性能,国内外研究者先后设计了多种算法[3-12],大致可分为包绕法、内外法、扫描法、分治法。随着应用的深入,大规模平面点集凸包的生成已成为 GIS 等领域面临的重要问题。栅格化方法具有可扩展性强、算法复杂性小、算法效率高等优点,其思想被广泛应用于 GIS 的相关算法中[22,23]。

1　栅格辅助的凸包生成算法

Alk 等[24]提出一种通过极值点分区的方法进行点集的预处理;Golin 等[25]利用矩形和斜线方程合并的方法取得凸包内的一个极大四边形,事先剔除该四边形内的点。这两种方法相较而言,前者易于实现且能有效地剔除凸包内的大部分点,被更为广泛地采用,如文献[26,27]中就采用该方法进行预处理。

为提高内点粗判效率,笔者借鉴在 GIS 中常用的栅格法思路,设计了一种栅格辅助的点集预处理方法,基本思路如下:建立一个能覆盖点集的栅格场,计算出每个点在栅格场中的位置(行号、列号),然后把每一行最左、最右栅格和每一列最上、最下栅格内的点保留下来,剔除点集中的其余点,仅用保留下来的点采用 Graham 扫描算法建立最小凸包。如图 1 所示,图中点集包含 100 个离散点,建立的栅格场为 10 行×10 列,需要保留的是以下格网单元(图中灰色格网)内的点:第 1 行的 D 和 G 列,第 2 行的 E 和 L 列,…,第 10 行的 C 和 F 列,以及 A 列的第 4 和 6 行,B 列的第 3 和 7 行,…,J 列的第 4 和 7 行。

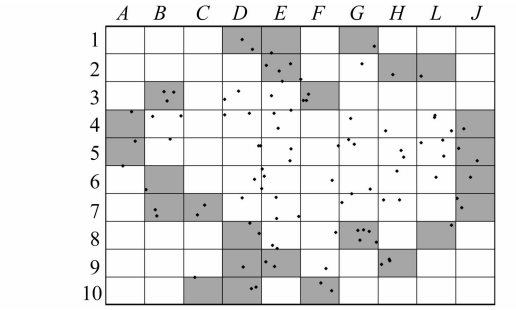

图 1　数据点集剔除过程

Fig. 1　Process of Rejecting Points Set

按上述思路设计的凸包生成算法,其完整步骤如下:① 计算点集的空间范围,确定栅格场参数(栅格大小、行列数等);② 定义 4 个一维数组 Left_Col、Right_Col、Top_Row、Bottom_Row,用以存储各行上有离散点分布的最左、最右列的序号,各列上有离散点分布的最上、最下行的序号;

收稿日期:2009-12-26。

项目来源:国家基础科学人才培养基金资助项目(J0630535)。

③ 扫描点集，计算并记录每个点在栅格场内的行列号，存储时可通过扩展点数据结构、定义独立的数组等方式实现，计算中及时更新步骤2中定义的数组；④ 提取满条件的点，对这些保留点采用Graham扫描算法建立凸包，提取条件为：判定点位于它所在行的最左（或最右）列，或所在列的最上（或最下）行。

图1中，当行列数均为1时，所有点都会被保留下来；行列数均很大时（如1 000行×1 000列），会导致每行上平均分布不足1点，点所在的列显然也是该行上存在数据点的最左、右列，根据规则也将被保留下来。由于点集中点的分布千差万别，从理论上计算具有最佳剔除效果的栅格行列数存在一定困难，这里仅进行一些初步分析。

设点集 S 的点数量为 N，点的分布相对均匀，栅格化时栅格总数为 M，行列数分别为 L 和 C，这里 $L\times C=M$。若将点集 S 视作一个完整的空间目标，而采用栅格辅助法剔除无关点的过程可看作是提取空间目标 S 的边界线（栅格形式），边界栅格占总栅格数比例 k 越小，则保留下来的点数越少，换言之，剔除无关点的效果越好。当 S 中点的分布近似于矩形时，待保留的边界栅格数为 $2(L+C-2)$，则栅格保留比 $k=2(L+C-2)/M\approx 2(L/M+1/L)$，当栅格总数 M 确定时，则该式中 $L=C=\sqrt{M}$ 时可得到 k 的最小值，此时栅格场的行列数相同。

设栅格总数为 M，行列数 $L=C=\sqrt{M}$，平均每个栅格的点数为 $n=N/M$，平均每行的点数为 $m=N/L=N/\sqrt{M}$。就单独一行而言，如图2所示，该行两端格网内的 $2n$ 个点需保留，$n\geqslant 1$ 时，保留比 $k=2n/m=2/\sqrt{M}$，该式表明 n 越小则保留比 k 越小；$n\leqslant 1$ 时，平均每个网格内分布不到1个点，由于行两端的点始终会被保留，因此，两端网格内的点数应计为2（不是 $2n$），此时 $k=2/m=2/\sqrt{M}$，n 取值越大则 k 越小。由此可见，当 $n=1$ 即平均每个网格有1个数据点时，保留比 k 最小，对无关点的剔除效果最佳，此时栅格场的行列数为 $L=C=\sqrt{N}$。

图2 单行格网的边界内点选取

Fig. 2 Select Boundary Points in Single Line of the Grid

为了评估上述观点，本文借助一些在特定图形限制下产生的随机点集进行了若干试验。图3为采用双椭圆限制产生的随机点集，测试点集中点的数量分别为50万、100万、250万以及500万，测试时栅格行数与列数相同，通过改变行数（或列数），测定采用栅格辅助法后可保留点的数量。测试结果如图4所示，图中横轴为栅格行列数，纵轴为原始点数与保留点数的比值，本处称之为剔除比，显然，纵轴方向取值越大，则表明剔除凸包内点越有效。

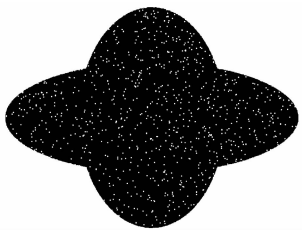

图3 双椭圆状测试点集

Fig. 3 Double Elliptical-shaped Point Set

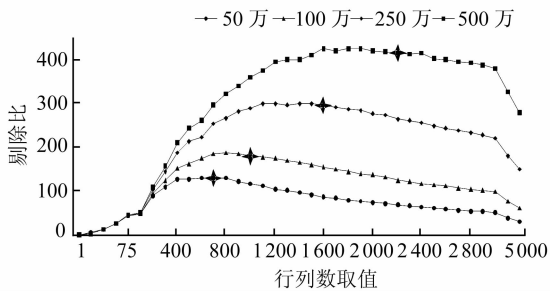

图4 栅格行列数与剔除比的关系测试结果(1)

Fig. 4 Test Result of Relationship Between Grid Row-column Number and Rejecting Proportion (1)

分析图4可知，尽管各组测试点集中点的数量不同，但“行列数-剔除比”曲线表现出相似的规律：随着行列数的增加，剔除比先快速递增，随后逐渐减小。考虑到制图效果，图中未绘制行列数超过5 000时的结果曲线，实际上其趋势仍是逐渐减小直至趋近于1。以其中50万点的测试点集为例，当行列数取2.5万、50万、75万时，剔除比分别为6.326、1.019、1.005。当栅格行列数取值为总点数的平方根时，即图4中以星号标记的位置，4条曲线上的剔除比均非常接近最大值（相差不超过5%），这与前面的理论分析相吻合。此外，图4的测试结果还表明，点集中点数越多，栅格辅助法的剔除效果越好，点数为500万时，剔除比达到400以上，换言之，99.75%的点被筛选出来不必参与后续的凸包计算。

图4的测试点集均是在双椭圆限制下产生的，点集的空间分布形态完全相同，或有其特殊性。为此，本文设计了另一组测试，包括3组空间

形态不同的数据，分别为椭圆形（图 5）、特殊图形（图 6）、长宽比为 10∶1 的长条形，每组均包含 36 万点。测试结果如图 7 所示，它所反映的规律与图 4 基本一致。

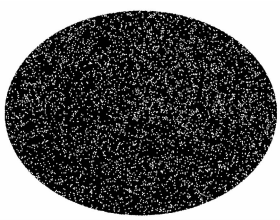

图 5　椭圆状测试点集
Fig. 5　Elliptical-shaped Point Set

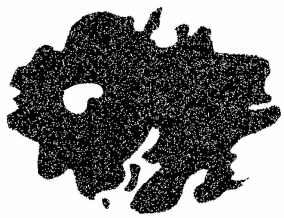

图 6　特殊形状测试点集
Fig. 6　Special-shaped Point Set

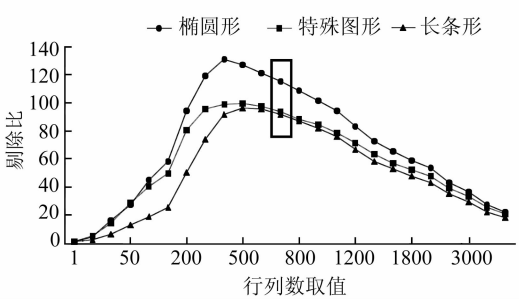

图 7　栅格行列数与剔除比的关系测试结果（2）
Fig. 7　Test Result of Relationship Between Grid Row-column Number and Rejecting Proportion (2)

2　测试与分析

本文算法的主要思想是借助栅格来判定和剔除那些对生成凸包没有贡献的内点，减少 Graham 扫描算法中参与排序的点数，以提高算法的总体性能。在预处理过程中，仅通过 3 次点集遍历即可实现内点的剔除，其时间复杂度为 $O(n)$，随后的 Graham 扫描算法时间复杂度为 $O(h\log h)$，其中 n 为点集中的总点数，h 为保留点的数量，当 $h \ll n$ 时总时间复杂度可近似达到 $O(n)$。

为评估算法的执行效率，本文采用双椭圆限制下产生的多组随机点集进行了测试，测试计算机配置为：Intel Core2 1.83 GHz，内存 1.5 GB，操作系统 Windows XP Pro SP3，编程环境为 Borland C^{++} Builder 6.0。测试中栅格行列数均设定为总点数的平方根，测试结果如表 1、图 8 所示。

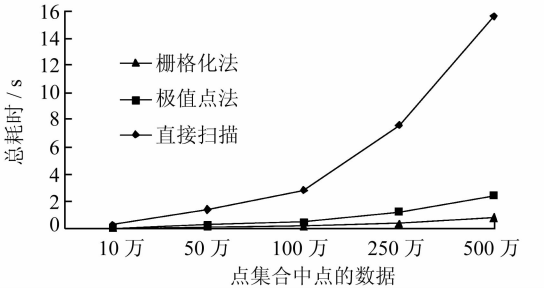

图 8　三种算法测试结果对比
Fig. 8　Comparison of Three Algorithms

表 1 的测试结果表明，采用栅格辅助法剔除部分内点后，算法效率的提高非常明显，与不采用预处理而直接通过 Graham 扫描算法生成凸包相比效率高了近 20 倍，与常用的通过极值点分区预处理方法相比效率提高 3 倍左右。在剔除内点的耗时方面，极值点分区法略高于栅格辅助法，因为前者需进行点与直线（极值点间的连线）位置关系的判断，计算量略高于“点的栅格化”；在剔除内点的数量上，栅格辅助法比极值点分区法高效得多，在点数为 500 万的测试中，前者剔除了 99.76% 的点，后者剔除了 88.47% 的点，前者保留下来参与凸包计算的点数仅为后者的 1/47。

表 1　测试结果表

Tab. 1　Test Results of Five Different Point Sets

总点数/万	栅格辅助法			极值点法			Graham 算法耗时/ms
	保留点数	预耗时/ms	总耗时/ms	保留点数	预耗时/ms	总耗时/ms	
10	1 649	14	15	11 725	16	31	234
50	3 773	62	78	58 282	93	218	1 344
100	5 534	141	156	116 186	156	437	2 828
250	8 533	360	375	288 727	406	1 156	7 562
500	12 037	719	750	576 266	828	2 406	15 594

注：预耗时指进行点集粗判、剔除无关点过程所耗时间，总耗时指从算法执行到计算出凸包的总时间。

仅就栅格辅助法本身而言，预耗时占用了绝大部分计算时间（80% 以上），这与剔除凸包内点的高效率相辅相成，其保留点数 h 已远小于总点数 n。在点数为 500 万的测试点集中 $h∶n$ 近似于 1∶415，此时计算凸包的总耗时随 n 的增加近似于线性变化，算法的时间复杂度可近似达到 $O(n)$。

本文借鉴 GIS 中栅格算法的思想，将离散点分配到相应的栅格块，仅使用边界栅格内的离散点进行最小凸包的计算，原始点集中的大部分点被事先剔除，从而有效地提高算法效率。在进一步研究中，可尝试将本文方法与分治法、极值点分

区等方法结合使用,例如对于参与凸包计算的保留点,进一步采用极值点分区法进行筛选;对分布不均的点集,单独针对密集区进行补充过滤。本文的有关探讨均是基于平面点集进行的,当GIS应用中涉及线、面等类型空间数据时需灵活处理。

参 考 文 献

[1] Thomas H C, Charles E L, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms, Second Edition [M]. MIT Press and McGraw-Hill, 2001: 955-956

[2] Böhm C, Kriegel H. Determining the Convex Hull in large Multidimensional Databases[C]. The Third International Conference on Data Warehousing and Knowledge Discovery, Lecture Notes in Computer Science, 2001: 294-306

[3] Chan T. Optimal Output-sensitive Convex Hull Algorithms in Two and Three Dimensions[J]. Discrete & Computational Geometry, 1996, 16: 361-368

[4] Bhattacharya B K, Sen S. On a Simple, Practical, Optimal, Output-sensitive Randomized Planar Convex Hull Algorithm[J]. Journal of Algorithms, 1997, 25(1):177-193

[5] 金文华, 何涛, 等. 简单快速的平面散乱点集凸包算法[J]. 北京航空航天大学学报, 1999, 25(1): 72-75

[6] 王杰臣. 二维空间数据最小凸包生成算法优化[J]. 测绘学报, 2002, 31(1): 82-86

[7] 王丽青, 陈正阳, 等. 一个改进的简单多边形凸包算法[J]. 计算机工程, 2007, 33(3): 200-201

[8] Jarvis R A. On the Identification of the Convex Hull of a Finite Set Points in the Plane[J]. Information Processing, 1973, 2:18-21

[9] 李成名, 陈军. Voronoi图生成的栅格算法[J]. 武汉测绘科技大学学报, 1998, 23(3), 208-210.

[10] 吴艳兰,胡鹏. 由栅格等高线快速建立DEM的新方法——CSE法[J]. 武汉大学学报·信息科学版, 2001, 26(1), 86-90

[11] Golin M, Sedgewick R. Analysis of a Simple Yet Efficient Convex Hull Algorithm[J]. Proceedings of the fourth annual symposium on Computational geometry, 1988,153-163

[12] 金文华,何涛, 等. 基于有序简单多边形的平面点集凸包快速求取算法[J]. 计算机学报, 1998, 21(6): 533-539

[13] Liu G H, Chen C B. A New Algorithm for Computing the Convex Hull of a Planar Point Set[J]. Journal of Zhejiang University Science A, 2007, 8(8):1 210-1 217

第一作者简介:王结臣,博士,副教授,现主要从事GIS理论与应用研究。

E-mail:wangjiechen@hotmail.com

A Gird-aided Algorithm for Determining the Minimum Convex Hull of Planar Points Set

WANG Jiechen[1] *CHEN Yanming*[1]

(1 Geographic Information Science Department, 49 Hankou Road, Nanjing University, Nanjing 210093, China)

Abstract: This paper presents a gird aided method for determining the convex hull of large amount data, the total time complexity of algorithm can achieve about $O(n)$. The basic idea of the method is as follows. First, build a gird field which can cover with all planar points; then, figure out the position of every point in the gird field (such as the row and column place in the field). Afterwards, reserve the points which are in the gird field of leftmost (or rightmost) column or top (or bottom) row. Finally, execute Graham's algorithm to generate the convex hull with reserved points. The result of the tests indicates that there has some relationship between grid parameters and algorithm efficiency. Such as, with the increase in row-column number, the computational efficiency first rapid increase, followed by a gradual decrease. Additionally, when the row-column number select the evolution of total planar points' number, the pretreatment effect of the algorithm almost reach maximum. Furthermore, with the increase of the number of planar points, the elimination efficiency of gird aided method improves.

Key words: minimum convex hull; rasterizing; algorithm

About the first author: WANG Jiechen, associate professor, majors in GIS development and application.

E-mail: wangjiechen@hotmail.com