

基于最大基线倾角智能逼近的凸壳新算法^{*})

周启海 黄涛 吴红玉 张元新

(西南财经大学经济信息工程学院 成都 610074)

摘要 本文评述了有代表性的折半分治递归凸壳算法,并利用同构化凸壳基本定理提出效率更高的最大倾角智能逼近凸壳新算法。本新算法的同构化特点是:1)找出给定二维点集最外点(指最左、最右、最高、最低点),即其X轴、Y轴坐标值最大、最小的四个初始极点;2)用该初始极点,把原二维点集分布域划分为四个子分布域;3)分别在这四个子分布域中,各基于自身最新所得极点依次动态构造其基线倾角最大的当前极点,并用这些极点作凸边,来逐步智能逼近和最终生成该给定二维点集的凸壳。

关键词 同构化,凸壳算法,分布域,最大倾角,智能逼近

A New Algorithm for Finding Convex Hull Based on Intelligent Approximating with a Maximum Pitch of Base Lines

ZHOU Qi-Hai HUANG Tao WU Hong-Yu ZHANG Yuan-Xin

(School of Economic Information Engineering, Southwestern University of Finance and Economics, Chengdu 610074)

Abstract In this paper, comment on a representative algorithm convex hull with half-dividing and recurrence; and a more efficient new algorithm to find a convex hull based on intelligent approximating with a maximum pitch is given by the isomorphic fundamental theorem of the convex hull. The isomorphic characters of the new algorithm are: 1) find out the outside-most poles which are the leftmost, rightmost, topmost and bottommost points on the convex hull, i. e. the four initial poles which have the maximum or the minimum coordinate value of the X or Y axis among all the points in given 2D point set; 2) divides the original distributed domain into four sub-domain with the initial poles; 3) in every sub-domain, constructs a current pole with a maximum pitch to its base line based on its last pole got just dynamically and sequentially, and draw the rims of this convex polygon with these poles for intelligent approximating for a convex hull of the given 2D point set step by step.

Keywords Isomorphic, Convex hull algorithm, Distributed domain, Pitch of base Lines, Intelligent approximating

1 引言

不少文献,都说明了研究、改进和提高二维点集或线段集凸壳(以下简称凸壳)算法及其效率的重要意义与许多工作^[1~18]。文[1]提出了可指导改进与优化凸壳算法的同构化凸壳基本定理;并依据同构化凸壳基本定理,阐明了“二维凸壳生成算法改进与优化的同构化方向”。在此,本文拟对另一种有代表性的折半分治凸壳算法进行评述,并基于同构化凸壳基本定理提出效率更高的最大倾角智能逼近凸壳新算法。

2 二维凸壳问题与凸壳算法描述

定义 1 设多边形 Q 的顶点是给定平面内的点 $Q_1(x_1, y_1), Q_2(x_2, y_2), \dots, Q_n(x_n, y_n)$ 。如果线段 $Q_i Q_j$ ($i \neq j, 1 \leq i \leq n, 1 \leq j \leq n$) 总不在多边形 Q 外,则称 Q 为凸多边形。

定义 2 设二维点集 $S = \{P_i(x_i, y_i) \mid 1 \leq i \leq m\}$ 由给定平面内的点构成。如果凸多边形 Q 顶点 $Q_i \in S$,且 Q 是覆盖 S 中各点的最小凸多边形,则称凸多边形 Q 为二维点集 S 的凸壳。

定义 3 如何寻求给定二维点集 $S = \{P_i(x_i, y_i) \mid 1 \leq i \leq m\}$ 的二维凸壳,称为二维凸壳问题。

定义 4 凡能构造性生成给定二维点集 $S = \{P_i(x_i, y_i) \mid 1 \leq i \leq m \geq 3\}$ 的二维凸壳的算法,统称二维凸壳生成算法。

3 二维凸壳生成算法研究评述

20 世纪 70 年代以来,二维凸壳所具有的问题复杂性与应用重要性,使国内外专家学者对凸壳算法颇为关注,并已提出诸如卷包裹凸壳算法、格雷厄姆凸壳算法、折半分治凸壳算法,等等^[5]。文[1]已对卷包裹凸壳算法、格雷厄姆凸壳算法作了基本评述。在此,本文仅对另一有代表性的折半分治凸壳算法作简要评述。

3.1 折半分治凸壳算法思想简介

文[2]说明了基于分治技术与递归方法的折半分治凸壳算法,其算法效率高于卷包裹凸壳算法、格雷厄姆凸壳算法,且总运行时间为 $t(m) = O(m \log m)$ 。

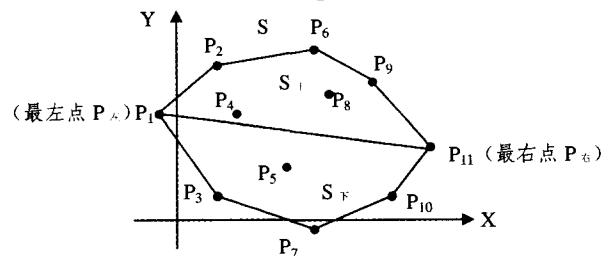


图 1 初始二维点集 S 划分为其上、下子点集 S_1, S_2 示意图

折半分治凸壳算法的算法思想,可简述如下。

^{*}) 基金项目:西南财经大学科研基金项目(No. 06K75)。周启海 教授,博(硕)士生导师,主要研究方向:计算几何,算法研究与实现,财经计算,同构化信息处理等。黄涛 讲师,主要研究方向:计算机应用。吴红玉 硕士研究生,主要研究方向:计算机应用。张元新 硕士研究生,主要研究方向:信息管理,计算机应用。

第0步:使初始二维点集 S 的点 $P_i(x_i, y_i)$, 满足 $x_i \leq x_{i+1}$, 其中 $1 \leq i \leq m-1, 3 \leq m < +\infty$. 并以初始点集 S 的最左点 $P_{\text{左}}$ 、最右点 $P_{\text{右}}$ 为端点的线段作上下分界线, 把 S 分为上下两个子点集 $S_{\text{上}}, S_{\text{下}}$ (例如如图 1 中: 当 $m=11$ 时, X 轴坐标值最小、最大点 P_1, P_{11} , 分别为初始点集 S 的最左点 $P_{\text{左}}$ 、最右点 $P_{\text{右}}$, 且除这两点 P_1, P_{11} 外, 还把点集 S 分为上子点集 $S_{\text{上}} = \{P_2, P_4, P_6, P_8, P_9\}$, 下子点集 $S_{\text{下}} = \{P_3, P_5, P_7, P_{10}\}$.)。

第 1 步: 调用递归算法 $UH(S_{\text{上}})$, 以生成上凸壳。

①如果当前 $m \leq 3$, 则当前点集 $S_{\text{上}}$ 中各点就是 $S_{\text{上}}$ 对应凸壳的各顶点, 并返回调用点; 否则, 执行②。

②按点数近似等量分治原则, 把当前点集 $S_{\text{上}}$ 折半划分为两个工作子点集 $S_{\text{上}1}, S_{\text{上}2}$ 。

③以 $S_{\text{上}1}$ 为新 $S_{\text{上}}$, 递归调用上凸壳生成递归算法 $UH(S_{\text{上}})$, 并生成 $S_{\text{上}1}$ 的凸壳 $Q_{\text{上}1}$ 。

④以 $S_{\text{上}2}$ 为新 $S_{\text{上}}$, 递归调用上凸壳生成递归算法 $UH(S_{\text{上}})$, 并生成 $S_{\text{上}2}$ 的凸壳 $Q_{\text{上}2}$ 。

⑤构造凸壳 $Q_{\text{上}1}$ 和凸壳 $Q_{\text{上}2}$ 的上公切线 (例如如图 2 所示的上虚线 Q_8Q_7), 把凸壳 $Q_{\text{上}1}$ 、凸壳 $Q_{\text{上}2}$ 合并为上凸壳 $Q_{\text{上}}$ 。

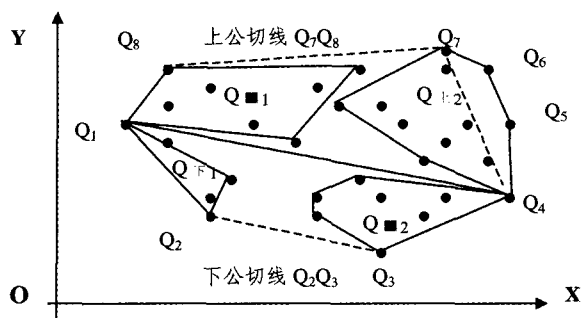


图 2 凸壳 Q 的“ $Q_{\text{上}1}, Q_{\text{上}2}$ 的上公切线, $Q_{\text{下}1}, Q_{\text{下}2}$ 的下公切线”示意图

第 2 步: 调用递归算法 $UH(S_{\text{下}})$, 以生成下凸壳。

①如果当前 $m \leq 3$, 则当前点集 $S_{\text{下}}$ 中各点就是 $S_{\text{下}}$ 对应凸壳的各顶点, 并返回调用点; 否则, 执行②。

②按点数近似等量分治原则, 把当前点集 $S_{\text{下}}$ 折半划分为两工作子点集 $S_{\text{下}1}, S_{\text{下}2}$ 。

③以 $S_{\text{下}1}$ 为新 $S_{\text{下}}$, 递归调用下凸壳生成递归算法 $UH(S_{\text{下}})$, 并生成 $S_{\text{下}1}$ 的凸壳 $Q_{\text{下}1}$ 。

④以 $S_{\text{下}2}$ 为新 $S_{\text{下}}$, 递归调用下凸壳生成递归算法 $UH(S_{\text{下}})$, 并生成 $S_{\text{下}2}$ 的凸壳 $Q_{\text{下}2}$ 。

⑤构造凸壳 $Q_{\text{下}1}$ 和凸壳 $Q_{\text{下}2}$ 的下公切线 (例如如图 2 所示下虚线 Q_2Q_3), 把凸壳 $Q_{\text{下}1}$ 、凸壳 $Q_{\text{下}2}$ 合并为下凸壳 $Q_{\text{下}}$ 。

第 3 步: 把上凸壳 $Q_{\text{上}}$ 与下凸壳 $Q_{\text{下}}$ 合而为一, 并生成凸壳 Q (例如如图 2 所示凸多边形 $Q_1Q_2Q_3Q_4Q_5Q_6Q_7Q_8$)。

3.2 折半分治凸壳算法主要弱点

不难看出, 折半分治凸壳算法存在的主要缺点是:

第一、由于采用递归, 故其算法效率不高。

第二、仅就上凸壳生成处理而言, 在“以上公切线 QQ_{i+1} 为一腰、上下分界线为另一腰, 而点 Q_i, Q_{i+1} 的 X 向坐标线为上、下底”所构成梯形 (实际上, 完全可扩大到: 在“以上公切线 QQ_{i+1} 为一边, 而以分界线 $P_{\text{左}}P_{\text{右}}$ 为其对边 $Q_{i+1}P_{\text{左}}; Q_{i+1}P_{\text{右}}$ 为另一边, 而以 $Q_iP_{\text{右}}$ 为其对边”所构成凸四边形 $P_{\text{左}}P_{\text{右}}Q_{i+1}Q_i$) 内, 若所包含的 $S_{\text{上}1}, S_{\text{上}2}$ 各自的点越多, 则该算法所作的子凸壳的无效边就越多, 因而会更降低其算法效率 (如图

2 所示: 非凸壳 Q 内的各无效边及其冗余处理, 就颇为不少.)。

4 最大倾角智能逼近凸壳算法

文 [1] 指出: 20 世纪提出的凸壳问题, 其算法研究始于 70 年代, 盛于 80 年代, 极于 90 年代; 然而, 进入 21 世纪以来, 传统的凸壳算法研究出现了停滞不前的尴尬窘况。为改变这种窘况, 权以本算法作为“引玉之砖”。

4.1 最大倾角智能逼近凸壳算法的理论基础

本算法的理论基础, 是同构化凸壳基本定理。

同构化凸壳基本定理 记二维点集 $S = \{P_i(x_i, y_i) \mid 1 \leq i \leq m, 3 \leq m < +\infty\}$, 凸多边形 Q 的顶点集为 $R = \{Q_k(x_k, y_k) \mid 1 \leq k \leq n \leq m < +\infty, 3 \leq n, 3 \leq m\}$ 。如果凸多边形 Q 是二维点集 S 的凸壳, 则有:

i. (顶点对内点的) 凸壳内点无关性定理: 凸壳 Q 的所有顶点 $Q_i(x_i, y_i)$ 必不在 S 与 R 的余集中, 即 $Q_i(x_i, y_i) \notin S - R, 1 \leq i \leq n \leq m < +\infty, 3 \leq n, 3 \leq m$ 。

ii. (顶点对顶点的) 凸壳顶点独立性定理: 凸壳 Q 的任一顶点 $Q_k(x_k, y_k)$ 必不在 R 与 $\{Q_k\}$ 的余集中, 即 $Q_k(x_k, y_k) \notin R - \{Q_k\}, 1 \leq k \leq n \leq m < +\infty, 3 \leq n$ 。

基于同构化凸壳基本定理, 作者认为凸壳生成算法改进与优化的同构化方向应当是: 第一、根据凸壳内点无关性定理, 应一方面使凸壳极点 (实为顶点) 分布域极小化, 即让包含凸壳极点的判定区域尽可能小, 以大大减少凸壳极点判定时的无效处理量; 另一方面使极点判定对象直接化, 即让所判定对象尽可能接近当前所寻极点, 以大幅提高凸壳极点判定对象的直接针对性。第二、根据凸壳顶点独立性定理, 一方面可从不同初始对象出发, 来改进和优化串行凸壳新算法; 另一方面可对不同视角对象处理, 来改造和创造并行凸壳新算法。因此, 在生成凸壳过程中, 应尽力缩小极点的可能分布域——在尽可能小的分布域内, 尽可能快地直接找出并只找出其各个极点 (即凸壳各条边的各端点) 的凸壳算法; 进而, 再对有潜力的优秀串行凸壳新算法施以并行化改造与创新。无疑, 这必定是今后进一步提高凸壳算法效率的主要捷径。

4.2 其基本定义与算法构造

下面简述依据同构化凸壳基本定理而提出的“基于最大倾角智能逼近的凸壳新算法”。

定义 5 二维点集 $S = \{P_i(x_i, y_i) \mid 1 \leq i \leq m \geq 3\}$ 中各点的位置分布区域, 称为 S 分布域。

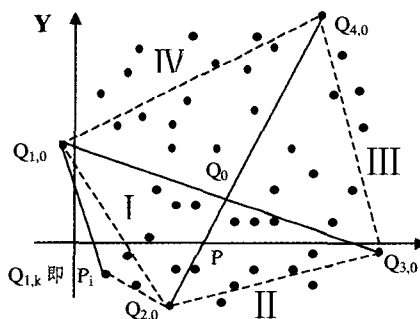


图 3 凸壳的初始极点、子分布域及其最大倾角智能逼近示意图

定义 6 二维点集 $S = \{P_i(x_i, y_i) \mid 1 \leq i \leq m \geq 3\}$ 中, 其 X 轴、 Y 轴坐标值最大、最小的四个最外点分别记为 $P_{(1)}(x_1$

$= \min\{x_i \mid 1 \leq i \leq m \geq 3\}, y_1), P_{(2)}(x_2, y_2 = \min\{y_i \mid 1 \leq i \leq m \geq 3\}), P_{(3)}(x_3 = \max\{x_i \mid 1 \leq i \leq m \geq 3\}, y_3), P_{(4)}(x_4, y_4 = \max\{y_i \mid 1 \leq i \leq m \geq 3\})$; 这四个最外点 $P_{(1)}, P_{(2)}, P_{(3)}, P_{(4)}$, 合称凸壳 Q 的初始极点, 并记为 $Q_{1,0}, Q_{2,0}, Q_{3,0}, Q_{4,0}$ 。线段 $Q_{1,0}Q_{3,0}$ 与线段 $Q_{2,0}Q_{4,0}$, 称为凸壳 Q 的基线, 两基线的交点 Q_0 称为凸壳 Q 的远心; 这两条基线所划分出的二维点集 S 分布域的四小区域 I、II、III、IV, 合称二维点集 S 的子分布域 (如图 3 所示)。

据此, 本文提出效率较高的最大倾角智能逼近凸壳算法。其算法思想可构造如下:

第 0 步: 初始化处理。

①“构造布域 S 的基线及其子分布域”处理: 生成凸壳 Q 的基线 $Q_{1,0}Q_{3,0}$ 与 $Q_{2,0}Q_{4,0}$, 并把初始二维点集 S 的原分布域划分为四个子分布域 I、II、III、IV。分别使各子分布域的点 $P_i(x_i, y_i)$, 各自满足 $x_i \leq x_{i+1}$ 。

②“分布域及其子分布域初始极小化”处理: 在分布域 S 中, 删除位于大基线四边形 $Q_{1,0}Q_{2,0}Q_{3,0}Q_{4,0}$ (注意: 它可能退化为三角形 $Q_{1,0}Q_{2,0}Q_{3,0}$) 中的所有内点; 并仍记为 S 。

第 1 步: 标记子分布域 I 为 S_1 。寻找凸壳 Q 在子分布域 S_1 中的各极点 (显然, 最紧邻两极点必顺次构成凸壳 Q 的一条边)。即

①“寻找下一新极点”处理: 当 S_1 中尚有未处理点时, 找出与当前子分布域 S_1 的基线倾角最大 (即满足 $\angle Q_0 Q_{1,0} P_i = \max\{\angle Q_0 Q_{1,0} Q_{i,1} \mid Q_{i,1} \in S_1\}$) 的最大倾角点 P_i (注: 若有多个最大倾角点, 则只取最后一个最大倾角点, 即离最值点 $Q_{1,0}$ 最远的那个倾角点); 顺次标记点 P_i 为 (即顶点) $Q_{1,k}, 1 \leq k$ 。

②“子分布域 S_1 极小化”处理: 在 S_1 中, 删除位于三角形 $Q_{1,0}P_iQ_{2,0}$ 中的所有内点; 并仍记为 S_1 。回到本步①。

第 2 步: 标记子分布域 II 为 S_1 。寻找凸壳 Q 在子分布域 S_1 中的各极点。即

①当 S_1 中尚有未处理点时, 找出与当前子分布域 S_1 的基线倾角最大 (即满足 $\angle Q_0 Q_{2,0} P_i = \max\{\angle Q_0 Q_{2,0} Q_{i,1} \mid Q_{i,1} \in S_1\}$) 的最大倾角点 P_i ; 顺次标记点 P_i 为新极点 $Q_{2,k}, 1 \leq k$ 。

②在 S_1 中, 删除位于三角形 $Q_{2,0}P_iQ_{3,0}$ 中的所有内点; 并仍记为 S_1 。回到本步①。

第 3 步: 标记子分布域 III 为 S_1 。寻找凸壳 Q 在子分布域 S_1 中的各极点。即

①当 S_1 中尚有未处理点时, 找出与当前子分布域 S_1 的基线倾角最大 (即满足 $\angle Q_0 Q_{3,0} P_i = \max\{\angle Q_0 Q_{3,0} Q_{i,1} \mid Q_{i,1} \in S_1\}$) 的最大倾角点 P_i ; 顺次标记点 P_i 为新极点 $Q_{3,k}, 1 \leq k$ 。

②在 S_1 中, 删除位于三角形 $Q_{3,0}P_iQ_{4,0}$ 中的所有内点; 并仍记为 S_1 。回到本步①。

第 4 步: 标记子分布域 IV 为 S_1 。寻找凸壳 Q 在子分布域 S_1 中的各极点。即

①当 S_1 中尚有未处理点时, 找出与当前子分布域 S_1 的基线倾角最大 (即满足 $\angle Q_0 Q_{4,0} P_i = \max\{\angle Q_0 Q_{4,0} Q_{i,1} \mid Q_{i,1} \in S_1\}$) 的最大倾角点 P_i ; 顺次标记点 P_i 为新极点 $Q_{4,k}, 1 \leq k$ 。

②在 S_1 中, 删除位于三角形 $Q_{4,0}P_iQ_{1,0}$ 中的所有内点; 并仍记为 S_1 。回到本步①。

最后, 顺序把分布域中 I、II、III、IV 所得各极点 (即顶点), 依次两两连接而得到的凸多边形 Q , 必定是所求二维有限点集 S 的凸壳 Q 。

4.3 本算法技术关键与核心基础的数学证明

基线倾角最大智能逼近凸壳算法的技术关键与核心基础, 是下列两个重要命题:

i. 凸壳 Q 的顶点, 与“对分布域及其子分布域的极小化

处理”无关。

ii. 基线倾角最大点 P_i , 必定是所求二维有限点集 S 的凸壳 Q 的一个极点 (即顶点)。

其数学证明如下:

i. 分布域及其子分布域的极小化处理中, 所删除的仅为 (且均为) 凸壳 Q 的内点, 故根据凸壳内点无关性定理与顶点独立性定理可知, 凸壳 Q 的各顶点必仍在极小化后的分布域及其子分布域中。

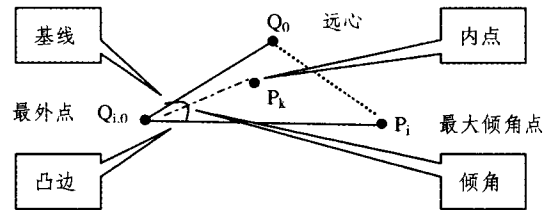


图 4 最大倾角智能逼近凸壳算法核心部分数学证明的示意图

ii. 如图 4 所示, 设凸壳 Q 的当前子分布域 S_i 的初始极点为 $Q_{i,0}$, 基线为 $Q_{i,0}Q_{j,0}$, 点 P_i 为 S_i 的最大角倾点。因 P_i 满足 $P_i = \max\{\text{基线倾角} \angle Q_{i,0}Q_{j,0}P_i \mid P_i \in S_i\}$, 且满足“若有多个最大倾角点, 则只取最后一个最大倾角点”; 故对 S_i 中任一点 P_k 必有 $\angle Q_{i,0}Q_{j,0}P_i \geq \angle Q_{i,0}Q_{j,0}P_k$, 即点 P_k 必在 $\triangle Q_{i,0}Q_{j,0}P_i$ 内, 或者至多同在凸边 $Q_{i,0}P_i$ 上 (但须注意: 此时, 两者长度必有 $Q_{i,0}P_k < Q_{i,0}P_i$)。因此, 点 P_i 必为凸壳 Q 的一个极点。证毕。

结论 本文提出的倾角最大智能逼近凸壳新算法, 不仅在时间、空间复杂度与效率上, 均显然优于现行卷包裹凸壳算法、格雷厄姆凸壳算法、折半分治凸壳算法等传统凸壳算法; 而且, 很容易改造为并行化算法。因此, 它将有效提高二维凸壳生成速度, 可进一步改进和提高二维凸壳在图像处理、文字分解、模式识别、物体分类、计算图形、指纹识别、遥测遥控、地物辨识、地质勘探、空天利用等的应用水平和工作效率。

参考文献

- 1 周启海. 二维点集或线段集凸壳生成算法改进与优化的同构化方向[J]. 计算机科学, 2007(6)
- 2 陈国良. 并行计算·结构·算法·编程[M]. 高等教育出版社, 2002
- 3 周培德. 计算几何·算法分析与设计[M]. 清华大学出版社, 2000
- 4 Aloupis G. A History of Linear-time Convex Hull Algorithms for Simple Polygons[J]. http://en.wikilib.com/wiki/Talk:Convex_hull
- 5 Sunday D. The Convex Hull of a 2D Point Set or Polygon[J]. <http://softsurfer.com/Archive/algorithm-0109/algorithm-0109.htm>
- 6 Rourke J O. Computational Geometry in C (2nd Edition), Chap. 3 "Convex Hulls in 2D"[M], 1998
- 7 Barber C, Dobkin D, Huhdanpaa H. The Quickhull algorithm for convex hulls[J]. ACM Trans. on Mathematical Software, 1997, 22: 469~483
- 8 Kirkpatrick D G, Seidel R. The Ultimate Planar Convex Hull Algorithm? [J]. SIAM Jour. Computer, 1986, 15: 287~299
- 9 Preparata F, Shamos M. Computational Geometry: An Introduction, Chap. 3 "Convex Hulls; Basic Algorithms"[M], 1985
- 10 Kallay M. The Complexity of Incremental Convex Hull Algorithms in R^d [J]. In: Proc. 1984, 19: 197
- 11 Andrew A M. Another Efficient Algorithm for Convex Hulls in Two Dimensions[J]. In: Proc. 1979, 9: 216~219
- 12 Akl S G, Toussaint G. Efficient Convex Hull Algorithms for Pattern Recognition Applications[J]. In: Proc. 4th Int'l Joint Conf. on Pattern Recognition[M]. Kyoto, Japan, 1978. 483~487
- 13 Bykat A. Convex Hull of a Finite Set of Points in Two Dimensions [J]. In: Proc. Letters 7, 1978. 296~298
- 14 Preparata F, Hong S J. Convex Hulls of Finite Sets of Points in Two and Three Dimensions[J]. Comm. ACM, 1977, 20: 87~93
- 15 Eddy W. A New Convex Hull Algorithm for Planar Sets[J]. ACM Trans. Math. Software, 1977, 3(4): 398~403
- 16 Jarvis R A. On the Identification of the Convex Hull of a Finite Set of Points in the Plane[J]. In: Proc. 1973, 2: 18~21
- 17 Graham R. An Efficient Algorithm for Determining the Convex Hull of a Finite Point Set[J]. In: Proc. 1972, 1: 132~133
- 18 Chand D, Kapur S. An algorithm for convex polytopes[J]. ACM, 1970, 17: 78~86