

基于四群四域四向动态基线倾角最大化圈绕的凸壳并行新算法

周启海 黄涛

(西南财经大学信息技术应用研究所 成都 610074) (西南财经大学经济信息工程学院 成都 610074)

摘要 首先把基线倾角最大化圈绕凸壳串行算法改进为动态基线倾角最大化圈绕凸壳串行算法;然后,根据同构化凸壳构造基本定理,利用工作站机群优点,进一步对动态基线倾角最大化圈绕凸壳串行算法施加多域化扩展与并行化改造,并提出效率更高的基于四群四域四向动态基线倾角最大化圈绕的凸壳并行新算法。该凸壳并行新算法的特点是:1)其机群分为4个子机群;2)其数据分布域分为4个子分布域;2)其各子分布域内凸壳顶点的圈绕寻找方向共有4个,即各子分布域均各由自己的逆时针寻找方向。

关键词 同构化,机群,凸壳,并行算法,四群,四域,四向

A New Parallel Algorithm for Finding Convex Hull Based on Maximum Pitch of the Dynamical Base Line Coiling with 4-Clusters, 4-Domains and 4-Directions

ZHOU Qi-Hai HUANG Tao

(Research Institute of Information Technology Application, Southwestern University of Finance and Economics, Chengdu 610074)

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

Abstract In this paper, the serial algorithm for finding convex hull based on maximum pitch of the base line coiling is improved to be the serial algorithm for finding convex hull based on maximum pitch of the dynamical base line coiling firstly; then, according to the isomorphic fundamental theorem of the convex hull construction and using the advantages of COW (Cluster of workstation), the serial algorithm for finding convex hull based on maximum pitch of the dynamical base line coiling is extended in multi-domains and is modified in parallel, further a more efficient new parallel algorithm to find a convex hull based on maximum pitch of the dynamical base line coiling with 4-Clusters, 4-Domains and 4-Directions is given. The general characters of the new parallel algorithm are, 1) its COW is combined with four sub-clusters; 2) its domain is divided into four sub-domains; 3) its seeking directions of coiling are four, which the seeking directions of coiling with a maximum pitch of the dynamical base line in every sub-domain is along with one way (anti clockwise direction) of itself separately.

Keywords Isomorphic, COW, Convex hull, Parallel algorithm, 4-Clusters, 4-Domains, 4-Directions

1 引言

始于20世纪70年代、盛于80年代、巛于90年代、滞于21世纪的二维点集凸壳(含线段集凸壳,以下简称凸壳,Convex hull)算法研究,迄今已近40年,已久无新进展。凸壳所具有的问题复杂性与应用重要性,引不少国内外专家学者竞相关注和研究凸壳算法,有不少文献论及凸壳算法改进和提高,已提出格雷厄姆凸壳算法等不少串行凸壳算法,与折半分治凸壳算法等并行凸壳算法^[1,2]。但总体上,这些凸壳算法效率往往还不够高(例如:不少并行凸壳算法,也多基于串行改造,且常采用递归方法),仍尚待进一步提高。为此,作者自2005年起潜心研究凸壳算法,并根据周启海教授提出的同构化凸壳构造基本定理^[1],陆续提出若干凸壳新算法^[3~8]。本文基于把文[4]给出的“最大基线倾角智能逼近凸壳串行算法”改进为“动态基线倾角最大化圈绕凸壳新算法”,并将后者作域化扩展与并行化改造,提出效率更高的基于工作站机群的四群四域四向基线倾角最大化圈绕的凸壳并行新算法。

2 二维凸壳问题与凸壳算法简述

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。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 顶点何寻求给定二维点集 $S = \{P_i \in S, \text{且 } Q \text{ 是可覆盖 } S \text{ 中各点的最小凸多边形,则称凸多边形 } Q \text{ 为二维点集 } S \text{ 的凸壳。如 } \{P_i(x_i, y_i) \mid 1 \leq i \leq m\}$ 的二维凸壳,称为二维凸壳问题。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。



A) 木版上各点处各钉一个图钉: B) 外围围绕各图钉所得橡皮带圈:
二维点集 S 凸壳 Q

图1 凸壳问题几何原型的形象说明示意图

凸壳问题的几何原型,可通俗简单而形象地说明如图1所示。设 $1 \leq i \leq m, 3 \leq m < +\infty$,对木版上的有限点集 S 中各点 $P_i(x_i, y_i)$ 处,分别各钉一个图钉,再用一条橡皮带从该

点集的外沿去围绕一圈这些图钉。显然有:首先,被缠紧的橡皮带圈必定构成一个凸多边形 Q ; 其次,所有图钉总不会出现在该橡皮带圈(即:所得凸多边形 Q)所围成的区域之外。

3 凸壳算法改进和优化的同构化方向与捷径

文[1]提出并阐明了下述同构化凸壳构造基本定理。

同构化凸壳构造基本定理 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 $R = \{Q_i(x_i, y_i) \mid 1 \leq i \leq n \leq m < +\infty, 3 \leq n, 3 \leq m\}$ 。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 $Q_k(x_k, y_k) \notin R - \{Q_k\}, 1 \leq k \leq n \leq m < +\infty, 3 \leq n$ 。

基于同构化凸壳构造基本定理,作者认为凸壳生成算法改进与优化的同构化方向应是:

第一、根据凸壳内点无关性定理,应一方面使凸壳顶点分布域极小化,即让包含凸壳顶点的判定区域尽可能小,以大大减少凸壳顶点判定时的无效处理量;另一方面使顶点判定对象直接化,即让所判定对象尽可能接近当前所寻顶点,以大幅提高凸壳顶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,应尽力缩小顶点的可能分布域——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顶点(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 动态基线倾角最大化围绕凸壳算法描述

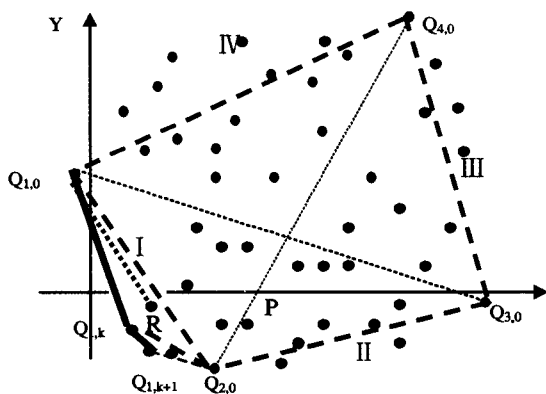


图2 凸壳的初始顶点、子分布域、动态基线倾角示意图

定义2 二维点集 $S = \{P_i(x_i, y_i) \mid 1 \leq i \leq m, 3 \leq m < +\infty\}$ 中各点的位置分布区域,称为 S 分布域。其 X 轴、 Y 轴坐标值最大、最小的四个最外点分别记为 $Q_{1,0}(x_1 = \min\{x_i \mid 1 \leq i \leq m, 3 \leq m < +\infty\}, y_1)$, $Q_{2,0}(x_2, y_2 = \min\{y_i \mid 1 \leq i \leq m, 3 \leq m < +\infty\})$, $Q_{3,0}(x_3 = \max\{x_i \mid 1 \leq i \leq m, 3 \leq m < +\infty\}, y_3)$, $Q_{4,0}(x_4, y_4 = \max\{y_i \mid 1 \leq i \leq m, 3 \leq m < +\infty\})$, 合称为二维点集 S 的凸壳 Q 的初始顶点,其中 $Q_{1,0}$ 可另记为 $Q_{5,0}$ 。

凸壳 Q 的初始顶点线段 $Q_{1,0}Q_{3,0}$ 、 $Q_{2,0}Q_{4,0}$ 均称为凸壳 Q 的直径;以凸壳直径为对角线的四边形 $Q_{1,0}Q_{2,0}Q_{3,0}Q_{4,0}$ (注意:它可能退化为三角形 $Q_{1,0}Q_{2,0}Q_{3,0}$),把二维点集 S 中的非四边形 $Q_{1,0}Q_{2,0}Q_{3,0}Q_{4,0}$ 内部点的所有点,沿逆时针划分为四个小区域 I、II、III、IV,均称为分布域 S 的子分布域 $S_i, i=1,2,3,4$ 。凸壳 Q 初始顶点线段 $Q_{1,0}Q_{2,0}$ 、 $Q_{2,0}Q_{3,0}$ 、 $Q_{3,0}Q_{4,0}$ 、 $Q_{4,0}Q_{1,0}$,均称为凸壳 Q 的初始动态基线。(如图2所示)

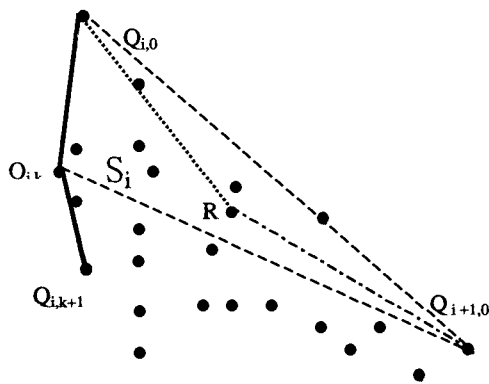


图3 子分布域 S_i 的基线与基线倾角示意图

定义3 为了叙述简便,初始顶点 $Q_{i+1,0}$ 可另记为 $Q_{i,-1}$ 。在子分布域 S_i 中,若点 $Q_{i,k}$ 是二维点集 S 的凸壳 Q 的顶点,点 $R \in S_i, 1 \leq i \leq 4, 0 \leq k$;则线段 $Q_{i,k}Q_{i,k-1}$ 称为子分布域 S_i 的动态基线,夹角 $\angle Q_{i,k-1}Q_{i,k}R$ 称为点 R 的动态基线倾角。(如图3所示)。

定理1 若点 $R \in$ 子分布域 S_i ,能使动态基线倾角 $\angle Q_{i,k-1}Q_{i,k}R = \max\{\angle Q_{i,k-1}Q_{i,k}X \mid X \in S_i\}$,则当前动态基线 $Q_{i,k-1}Q_{i,k}$ 的最大倾角点 R (注:若有多个最大倾角点,则只取离当前最大倾角点 $Q_{i,k-1}$ 最远的那个最大倾角点)必为 S_i 的子凸壳 Q_S 的下一紧邻顶点 $Q_{i,k+1}$ 。

可用反证法证明定理1如下:已知点 $R \in$ 子分布域 S_i ,是满足定理1条件中当前动态基线 $Q_{i,k-1}Q_{i,k}$ 的最大倾角点。假设点 R 不是子分布域 S_i 的子凸壳 Q_S 的下一紧邻顶点,即 R 只是子分布域 S_i 的子凸壳 Q_S 的一个内点或非下一紧邻顶点,故子凸壳 Q_S 的当前动态基线最大倾角点 $Q_{i,k}$ 必有其下一紧邻顶点 $P \in$ 子分布域 S_i ,且必在三角形 $\triangle Q_{i,k-1}Q_{i,k}R$ 外部(否则,点 P 就是 $\triangle Q_{i,k-1}Q_{i,k}R$ 中的点,即为凸壳 Q_S 的内点,从而不能作为凸壳 Q_S 的顶点)。此时:1)若当前动态基线最大倾角点 $Q_{i,k}$ (注意:它就是凸壳 Q_S 的当前顶点)的下一紧邻顶点 P ,是与 $\triangle Q_{i,k-1}Q_{i,k}R$ 同在直线 $Q_{i,k}R$ 的一侧,则 $\angle Q_{i,k-1}Q_{i,k}P < \angle Q_{i,k-1}Q_{i,k}R = \max\{\angle Q_{i,k-1}Q_{i,k}X \mid X \in S_i\}$,故至少必有一顶点 W 与顶点 P 各在直线 $Q_{i,k}R$ 的两侧,而顶点 W 显然比顶点 P 更临近当前最大倾角点 $Q_{i,k}$ 。这表明点 P 并不是当前最大倾角点 $Q_{i,k}$ 的下一紧邻顶点,从而与“ P 是当前最大倾角点 $Q_{i,k}$ 的下一紧邻顶点”相矛盾。2)若当前动态基线最大倾角点 $Q_{i,k}$ 的下一紧邻顶点 P 是与 $\triangle Q_{i,k-1}Q_{i,k}R$ 不在直线 $Q_{i,k}R$ 的同侧,则 $\angle Q_{i,k-1}Q_{i,k}P > \angle Q_{i,k-1}Q_{i,k}R$;但这与 $\angle Q_{i,k-1}Q_{i,k}R = \max\{\angle Q_{i,k-1}Q_{i,k}X \mid X \in S_i\} \geq \angle Q_{i,k-1}Q_{i,k}P$ 相矛盾。因此,定理1成立。

据此,作者把文[4]给出的“基于最大倾角智能逼近的凸壳算法”,改进为下述“动态基线倾角最大化围绕凸壳新算法”。

第0步;初始化处理。“构造分布域 S 的初始顶点”处理:二维点集 $S = \{P_i(x_i, y_i) \mid 1 \leq i \leq m, 3 \leq m < +\infty\}$ 中,找出

其 X 轴、Y 轴坐标值最大、最小的四个最外点,作为二维点集 S 的凸壳 Q 的初始顶点,并记为 $Q_{1,0}$ (它另记为 $Q_{5,0}$)、 $Q_{2,0}$ 、 $Q_{3,0}$ 、 $Q_{4,0}$ (其中: X 轴坐标值最小的点记为 $Q_{1,0}$)。删除以凸壳 Q 两条直径为对角线的四边形 $Q_{1,0}Q_{2,0}Q_{3,0}Q_{4,0}$ (注意: 它可能退化为三角形 $Q_{1,0}Q_{2,0}Q_{3,0}$) 中各内点,所得二维点集仍记为 S_0 ; 沿逆时针把当前二维点集 S 分布域分划为子分布区域 $S_i, i=1,2,3,4$ 。

第 1 步: 标记子分布域 S_1 为 S_0 。以 $Q_{1,0}$ 为初始顶点,以 $Q_{1,0}Q_{2,0}$ (即 $Q_{1,0}Q_{1,-1}$, 下类同) 为初始动态基线, 寻找凸壳 Q 在子分布域 S_0 中的各顶点 (显然, 最紧邻两顶点必顺次构成凸壳 Q 的一条边)。即

①“寻找下一新顶点”处理: 当 S_0 中尚有未处理点 $R \in S_0$ 时, 找出点 R 与当前子分布域 S_0 中的动态基线 $Q_{1,k}Q_{1,k-1}$ 倾角最大 (即满足 $\angle Q_{1,k+1}Q_{1,k}Q_{1,k-1} = \max\{\angle RQ_{1,k}Q_{1,k-1} | R \in S_0\}$) 的最大倾角点 $Q_{1,k+1}$ (注: 若有多个最大倾角点, 则只取最远处的那个最大倾角点, 即离当前最大倾角顶点 $Q_{1,k}$ 最远的那个最大倾角点); 顺次标记新顶点 $Q_{1,k+1}, 0 \leq k$ 。

②“子分布域 S_0 极小化”处理: 在 S_0 中, 删除位于当前动态基线三角形 $Q_{1,k+1}Q_{1,k}Q_{1,k-1}$ 中的所有内点, 并仍记为 S_0 ; 以 $Q_{1,k+1}Q_{1,k}$ 为求下一顶点的新动态基线, 并仍记为 $Q_{1,k}$ 。回到第 1 步①。

第 2 步: 标记子分布域 S_2 为 S_0 。以 $Q_{2,0}$ 为初始顶点, 以 $Q_{2,0}Q_{3,0}$ (即 $Q_{2,0}Q_{2,-1}$, 下类同) 为初始动态基线, 寻找凸壳 Q 在子分布域 S_0 中的各顶点 (显然, 最紧邻两顶点必顺次构成凸壳 Q 的一条边)。即

①“寻找下一新顶点”处理: 当 S_0 中尚有未处理点 $R \in S_0$ 时, 找出点 R 与当前子分布域 S_0 中的动态基线 $Q_{2,k}Q_{2,k-1}$ 倾角最大 (即满足 $\angle Q_{2,k+1}Q_{2,k}Q_{2,k-1} = \max\{\angle RQ_{2,k}Q_{2,k-1} | R \in S_0\}$) 的最大倾角点 $Q_{2,k+1}$ (注: 若有多个最大倾角点, 则只取最远处的那个最大倾角点, 即离当前最大倾角顶点 $Q_{2,k}$ 最远的那个最大倾角点); 顺次标记新顶点 $Q_{2,k+1}, 0 \leq k$ 。

②“子分布域 S_0 极小化”处理: 在 S_0 中, 删除位于当前动态基线三角形 $Q_{2,k+1}Q_{2,k}Q_{2,k-1}$ 中的所有内点, 并仍记为 S_0 ; 以 $Q_{2,k+1}Q_{2,k}$ 为求下一顶点的新动态基线, 并仍记为 $Q_{2,k}$ 。回到第 2 步①。

第 3 步: 标记子分布域 S_3 为 S_0 。以 $Q_{3,0}$ 为初始顶点, 以 $Q_{3,0}Q_{4,0}$ (即 $Q_{3,0}Q_{3,-1}$, 下类同) 为初始动态基线, 寻找凸壳 Q 在子分布域 S_0 中的各顶点 (显然, 最紧邻两顶点必顺次构成凸壳 Q 的一条边)。即

①“寻找下一新顶点”处理: 当 S_0 中尚有未处理点 $R \in S_0$ 时, 找出点 R 与当前子分布域 S_0 中的动态基线 $Q_{3,k}Q_{3,k-1}$ 倾角最大 (即满足 $\angle Q_{3,k+1}Q_{3,k}Q_{3,k-1} = \max\{\angle RQ_{3,k}Q_{3,k-1} | R \in S_0\}$) 的最大倾角点 $Q_{3,k+1}$ (注: 若有多个最大倾角点, 则只取最远处的那个最大倾角点, 即离当前最大倾角顶点 $Q_{3,k}$ 最远的那个最大倾角点); 顺次标记新顶点 $Q_{3,k+1}, 0 \leq k$ 。

②“子分布域 S_0 极小化”处理: 在 S_0 中, 删除位于当前动态基线三角形 $Q_{3,k+1}Q_{3,k}Q_{3,k-1}$ 中的所有内点, 并仍记为 S_0 ; 以 $Q_{3,k+1}Q_{3,k}$ 为求下一顶点的新动态基线, 并仍记为 $Q_{3,k}$ 。回到第 3 步①。

第 4 步: 标记子分布域 S_4 为 S_0 。以 $Q_{4,0}$ 为初始顶点, 以 $Q_{4,0}Q_{5,0}$ (即 $Q_{4,0}Q_{4,-1}$, 下类同) 为初始动态基线, 寻找凸壳 Q 在子分布域 S_0 中的各顶点 (显然, 最紧邻两顶点必顺次构成凸壳 Q 的一条边)。即

①“寻找下一新顶点”处理: 当 S_0 中尚有未处理点 $R \in$

S_0 时, 找出点 R 与当前子分布域 S_0 中的动态基线 $Q_{4,k}Q_{4,k-1}$ 倾角最大 (即满足 $\angle Q_{4,k+1}Q_{4,k}Q_{4,k-1} = \max\{\angle RQ_{4,k}Q_{4,k-1} | R \in S_0\}$) 的最大倾角点 $Q_{4,k+1}$ (注: 若有多个最大倾角点, 则只取最远处的那个最大倾角点, 即离当前最大倾角顶点 $Q_{4,k}$ 最远的那个最大倾角点); 顺次标记新顶点 $Q_{4,k+1}, 0 \leq k$ 。

②“子分布域 S_0 极小化”处理: 在 S_0 中, 删除位于当前动态基线三角形 $Q_{4,k+1}Q_{4,k}Q_{4,k-1}$ 中的所有内点, 并仍记为 S_0 ; 以 $Q_{4,k+1}Q_{4,k}$ 为求下一顶点的新动态基线, 并仍记为 $Q_{4,k}$ 。回到第 4 步①。

最后步: 形成凸壳处理: 顺序把子分布域 S_i 中所得各顶点, 依次两两连接而得到的凸多边形 Q, 必定是所求原二维有限点集 S 的凸壳 Q。

显然, 本文提出的“动态基线倾角最大化围绕凸壳新算法”, 很容易改造为并行化算法。

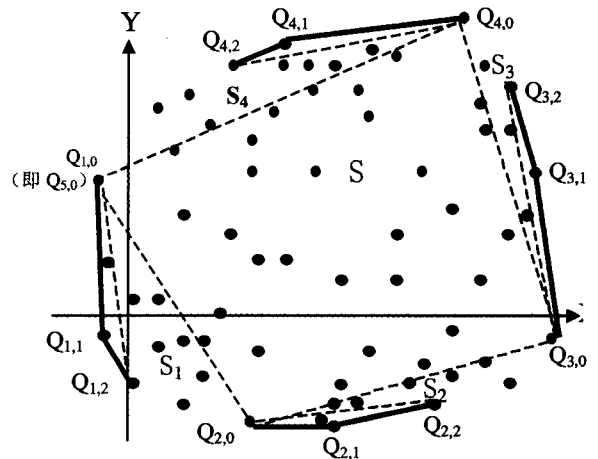


图 4 凸壳的各子分布域、初始顶点、动态基线倾角示意

5 四群四域四向动态基线倾角最大化围绕凸壳并行新算法

2007 年, 笔者依据同构化凸壳构造基本定理, 对“动态基线倾角最大化围绕凸壳串行算法”加以多域化扩展与并行化改造, 构造出效率更高的基于工作站机群的四群四域四向动态基线倾角最大化围绕并行凸壳新算法 (如图 4 所示)。其主要算法思想, 可简要概述如下:

第 0 步: 并行初始化处理。

(1)“寻找二维点集 S 的分布域 (仍记为 S) 的 X 轴、Y 轴坐标值最大、最小的 4 个最外点”的 4 群并行处理:

1) 设定 (注: 可由用户自行决定并输入) 初始分布域 $S = \{P_k(x_k, y_k) | 1 \leq k \leq m \geq 3\}$ 中各点的 X 轴坐标取值的最大、最小可能值分别为 X_{\max} 、 X_{\min} , Y 轴坐标取值的最大、最小可能值分别为 Y_{\max} 、 Y_{\min} 。

2) 分别标记构成所论机群 COW 的 4 个子机群为 $COW_i (i=1,2,3,4)$; 标记子机群 COW_i 下属各处理机的总数为 n_i ; 标记子机群 COW_i 下属各处理机为 $P_{ij} (1 \leq j \leq n_i)$ 。

3) 并行地使子机群 COW_i 下属各处理机 $P_{i,j} (i=1,2,3,4, 1 \leq j \leq n_i)$, 各用初始分布域 S 的 X 轴坐标可能值初始值域 $[X_{\min}, X_{\max}]$ 的初始带宽 $W_S = (X_{\max} - X_{\min}) / (n_1 + n_2 + n_3 + n_4)$, 对初始分布域 S 作带状划分出子分布域 $S_{i,j}$, 并各自保存所获得的初始子分布域 $S_{i,j}$ 内各点; 然后各对自己的初始子分布域 $S_{i,j}$:

①如果初始子分布域 $S_{i,j}$ 非空, 则使其处理机 $P_{i,j}$ 在自己

初始子分布域 $S_{i,j}$ 内,并行找各自的 X 轴、 Y 轴坐标值最大、最小的 4 个最外点 $(x_{i,j,\max} = \max\{x \mid (x,y) \in S_{i,j}\}, y_{右,i,j})$ 、 $(x_{i,j,\min} = \min\{x \mid (x,y) \in S_{i,j}\}, y_{左,i,j})$ 、 $(x_{高,i,j}, y_{i,j,\max} = \max\{y \mid (x,y) \in S_{i,j}\})$ 、 $(x_{低,i,j}, y_{i,j,\min} = \min\{y \mid (x,y) \in S_{i,j}\})$;

②从各处理机 $P_{i,j}$ 所得的各初始子分布域 $S_{i,j}$ ($1 \leq j \leq n_i$) 的全部最右点集 $R_{ight} = \{(x_{i,j,\max}, y_{右,i,j}) \mid 1 \leq j \leq n_i\}$ 、全部最左点集 $L_{eft} = \{(x_{i,j,\min}, y_{左,i,j}) \mid 1 \leq j \leq n_i\}$ 、全部最高点集 $H_{igh} = \{(x_{高,i,j}, y_{i,j,\max}) \mid 1 \leq j \leq n_i\}$ 、全部最低点集 $L_{ow} = \{(x_{低,i,j}, y_{i,j,\min}) \mid 1 \leq j \leq n_i\}$ 中,并行找出各初始子分布域 S_i 的 4 个最外点:最右点 $(x_{i,\max} = \max\{x \mid (x,y) \in R_{ight}\}, y_{右,i})$ 、最左点 $(x_{i,\min} = \min\{x \mid (x,y) \in L_{eft}\}, y_{左,i})$ 、最高点 $(x_{高,i}, y_{i,\max} = \max\{y \mid (x,y) \in H_{igh}\})$ 、最低点 $(x_{低,i}, y_{i,\min} = \min\{y \mid (x,y) \in L_{ow}\})$;

③从各子机群 COW_i ($1 \leq i \leq 4$) 的全部最外点集 $\{(x_{i,\max}, y_{右,i}) \mid 1 \leq i \leq 4\}$ 、 $\{(x_{i,\min}, y_{左,i}) \mid 1 \leq i \leq 4\}$ 、 $\{(x_{高,i}, y_{i,\max}) \mid 1 \leq i \leq 4\}$ 、 $\{(x_{低,i}, y_{i,\min}) \mid 1 \leq i \leq 4\}$ 中,分别找出初始子分布域 S 的 4 个(注意:它可能退化为 3 个)最外点,作为并标记为二维点集 S 的凸壳 Q 的初始顶点 $Q_{1,0}(x_{\min,1}, y_{0,\min,1})$ 、 $Q_{2,0}(x_{0,\min,2}, y_{\min,2})$ 、 $Q_{3,0}(x_{\max,3}, y_{0,\max,3})$ 、 $Q_{4,0}(x_{0,\max,4}, y_{\max,4})$;为叙述简洁,其初始顶点 $Q_{1,0}(x_{\min,1}, y_{0,\min,1})$ 也可另记为 $Q_{5,0}(x_{\min,5}, y_{0,\min,5})$ 。

4)各子机群 COW_i ($1 \leq i \leq 4$) 并行地从分布域 S 中删除其最外点四边形 $Q_{1,0}Q_{2,0}Q_{3,0}Q_{4,0}$ 的所有点(注:它除 4 个顶点外,其余各点均为凸壳 Q 的内点),所剩分布域仍记为 S 。

(2)“构造分布域 S 的 4 个子分布域 S_1, S_2, S_3, S_4 (注意:它可能退化为 3 个子分布域 S_1, S_2, S_3)”的 4 群并行处理:

1)各子机群 COW_i ($1 \leq i \leq 4$) 并行地对初始分布域 S ,按照实际值初始值域 $[x_{\min,1}, x_{\max,3}]$ 的实际带宽 $W_S = (x_{\max,3} - x_{\min,1}) / (n_1 + n_2 + n_3 + n_4)$,带状划分出子分布域 $S_{i,j}$ 。

2)并行地分别使子机群 COW_i ($1 \leq i \leq 4$),以由初始顶点 $Q_{1,0}, Q_{i+1,0}$ 构成的各自初始分布域 S_i 的动态基线 $Q_{i,0}Q_{i+1,0}$ 为分界判据,把“点 $P(x,y) \in S$,且与最外点四边形 $Q_{1,0}Q_{2,0}Q_{3,0}Q_{4,0}$ 位于动态基线 $Q_{i,0}Q_{i+1,0}$ 不同侧”的所有点,保存为供后续并行处理的二维点集 S 分布域的子分布域 S_i (注意: $S_1 = S_5$),并删除位于最外点四边形 $Q_{1,0}Q_{2,0}Q_{3,0}Q_{4,0}$ 中的所有内点。

第 1 步:“子机群 COW_i ($1 \leq i \leq 4$) 分别在子分布域 S_i 中生成子凸壳 Q_i 各顶点”的并行化处理。

第 1-1 步 子机群 COW_i 在子分布域 S_i 中,进行动态基线(逆时针方向;下同,略)倾角最大化圈绕寻找子凸壳 Q_i 各顶点的并行处理。

第 1-1-1 步 “子机群 COW_i 动态基线倾角最大化圈绕寻找子凸壳 Q_i 下一新顶点”的并行处理:

第 1-1-1-1 步 初始子分布域 S_i 仍标记为当前子分布域 S_i ;连接初始顶点 $Q_{1,0}, Q_{2,0}$ 构成当前动态基线 $Q_{1,0}Q_{2,0}$;置当前新顶点的序号计数器 r 初值为 0。

第 1-1-1-2 步 并行地使子机群 COW_i 下属各处理机 $P_{i,j}$ ($1 \leq j \leq n_i$),用带宽 $W_i = (\max\{x \mid (x,y) \in S_i\} - \min\{x \mid (x,y) \in S_i\}) / n_i$,把当前子分布域 S_i 带状划分为 n_i 个子分布域 $S_{i,j}$ ($1 \leq j \leq n_i$)。

第 1-1-1-3 步 并行地使子机群 COW_i 下属各处理机 $P_{i,j}$ 各在自己的子分布域 $S_{i,j}$ ($1 \leq j \leq n_i$) 的全部点中,均以 $Q_{i,r}$ 为子凸壳 Q_i 的次新顶点,求出其动态基线倾角最大化圈

绕的动态基线倾角最大点 $R_{i,j,0}$ (注:若有多个最大倾角点,则只取最远处的那个最大倾角点,即离当前最大倾角顶点 $Q_{i,r}$ 最远的那个最大倾角点);在所得各组动态基线倾角最大点集 $\{R_{i,j,0} \mid 1 \leq j \leq n_i\}$ 中,找出当前子分布域 S_i 的动态基线倾角最大点 R ,并标记为子凸壳 Q_i 的最新顶点 $Q_{i,r+1}$;置序号计数器 r 为 $r+1$ 。

第 1-1-2 步 “删除由子凸壳 Q_i 的初始顶点 $Q_{i,0}$,当前次新顶点 $Q_{i,r-1}$ 、最新顶点 $Q_{i,r}$ 所构成三角形 $Q_{i,0}Q_{i,r-1}Q_{i,r}$ 的所有点”的子分布域 S_i 极小化并行处理:

第 1-1-2-1 步 并行地使子机群 COW_i 下属各处理机 $P_{i,j}$ 各自生成“子凸壳 Q_i 的初始顶点 $Q_{i,0}$,当前次新顶点 $Q_{i,r-1}$ 、最新顶点 $Q_{i,r}$ ”所构成的三角形 $Q_{i,0}Q_{i,r-1}Q_{i,r}$,并记为 Δ_i 。

第 1-1-2-2 步 并行地使子机群 COW_i 下属各处理机 $P_{i,j}$ ($1 \leq j \leq n_i$),用带宽 $W_i = (\max\{x \mid (x,y) \in \Delta_i\} - \min\{x \mid (x,y) \in \Delta_i\}) / n_i$,把当前 Δ_i 带状划分为 n_i 个子分布域 $\Delta_{i,j}$ ($1 \leq j \leq n_i$)。

第 1-1-2-3 步 并行地使子机群 COW_i 下属各处理机 $P_{i,j}$ ($1 \leq j \leq n_i$),各自删除在自己的子区域 $\Delta_{i,j}$ ($1 \leq j \leq n_i$) 中的全部点。

第 1-1-2-4 步 把删除 Δ_i 中全部点后的原子分布域 S_i ,仍标记为当前子分布域 S_i ;若当前子分布域 S_i 非空(即还有未找出的凸壳顶点),则把子凸壳 Q_i 的初始顶点 $Q_{i,0}$ 、最新顶点 $Q_{i,r}$ 记为新的动态基线 $Q_{i,0}Q_{i,r}$,回到第 1-1-1-2 步,否则转而执行第 1-1-3 步。

第 1-1-3 步 子凸壳 Q_i 的全部顶点标记处理:顺次标记已求得的子凸壳 Q_i 全部顶点。

最后步:形成凸壳。顺序把各子分布域 S_1, S_2, S_3, S_4 中所得各顶点,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 。

结论 显而易见,本文提出的基于机群的四群四域四向动态基线倾角最大化圈绕凸壳并行新算法,利用机群技术实现了从双群扩大为四群(故其处理机个数倍增),而分布域从双域扩展为四域(故其子分布域更加小域化)。该算法与双群双域四向水平倾角最小化圈绕凸壳并行新算法^[8]相比,可进一步改进和提高其算法效率,且更易于推广到基于机群的 m 群、 n 域、 p 向($m > 2, n > 2, p > 2$) 的凸壳并行新算法研究。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. 论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, 34(7)
- 2 周启海. 简论二维点集凸壳研究的意义、现状与创新[C]. 见: 第三届全国几何设计与计算学术会议论文集, 北京: 电子工业出版社, 2007
- 3 周启海, 杨祥茂, 吴红玉. 单域单向水平倾角最小化圈绕凸壳新算法[J]. 西华大学学报(自科版), 2006(2)
- 4 周启海, 黄涛, 吴红玉, 张元新. 基于最大基线倾角智能逼近的凸壳新算法[J]. 计算机科学, 2007, 34(9)
- 5 周启海, 吴红玉, 黄涛. 单域双向水平倾角最小化圈绕凸壳新算法[J]. 计算机科学, 2007, 34(8)
- 6 黄涛, 周启海. 双域单向水平倾角最小化圈绕凸壳新算法[J]. 计算机科学, 2007, 34(12)
- 7 黄涛, 周启海, 吴红玉. 双域双向水平倾角最小化圈绕凸壳新算法[J]. 计算机科学, 2008, 35(1)
- 8 周启海, 黄涛. 基于双群双域四向水平倾角最小化圈绕的凸壳并行新算法[J]. 计算机科学, 2008, 35(1)