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*)

周启海

(西南财经大学经济信息工程学院　成都 610074)

摘　要　本文指出了迄今为止的现行二维点集或线段集(包括:多边形、封闭折线、半封闭折线、开放线段集等)凸壳生成算法的共同弱点;提出了可改进与优化凸壳算法的同构化凸壳构造基本定理。进而,基于同构化凸壳构造基本定理,阐明了有限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,应当是:第一,使凸壳极点(或称顶点)分布域极小化,即让包含凸壳极点的判定区域尽可能小;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。第二,尽力对有可改造潜力的优秀串行凸壳算法施以并行化改造和创新。

关键词　凸壳算法,同构化凸壳构造基本定理,分布域极小化,判定对象直接化

On an Isomorphic Direction of Improving and Optimizing an Algorithm for Determining the Convex Hull of 2D Point Set or Line Segment Set

ZHOU Qi-Hai

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

Abstract　In this paper, the common weakness of the current algorithms for seeking the convex hull of a finite 2D points set or Line Segment Set (include: polygons, closure broken lines, semi-closure broken lines, opening line segments, etc.) is pointed out up to day; the isomorphic fundamental theorem of the convex hull is raised, with which an algorithm for determining the convex hull would be improved and optimized. Further, based on the isomorphic fundamental theorem of the convex hull, an isomorphic direction of improving and optimizing an algorithm for determining the convex hull of a finite 2D points set or line segments set is clarified, which should be: 1). a minimum distributed domain of the poles of the convex hull, i. e. , make that the domain including all poles of the convex hull is as small as possible; a immediate objects-judged for all poles of the convex hull, i. e. , make all objects-judged are as near by the current poles of the convex hull to be determined as possible. 2) try one's best to improve the excellent sequential algorithms, which would be reformed, for finding the convex hull into their parallel algorithms, or create other new parallel algorithms.

Keywords　Convex hull algorithm, Isomorphic fundamental theorem of the convex hull, Minimum distributed domain, Immediate objects-judged

1　引言

二维凸壳即"可覆盖二维有限点集或线段集(包括:多边形、封闭折线、半封闭折线、开放线段集)的最小外接凸多边形"(以下简称凸壳),有重大理论研究和实际应用价值。例如:计算机图形学中,可用一组点的凸壳,显示出其点簇(cluster of points);图像处理中,可用寻求图像凸壳,找到数字图像中的关键凸面;指纹识别中,可据指纹边缘轮廓点集凸壳,获得高质量的指纹;模式识别中,可借模式凸壳,描述模式外形的重要特征;空天地面视物辨识中,可用凸壳快速识别地面场景区域;古繁体文字分解中,可以构造字形凸包,形成对文字的最优划分;物体分类中,可凭各物体凸壳相似度,勾画出这些物体所属类别;等等。显然,研究、改进和提高凸壳生成算法及其效率有重要意义[1~4]。为此,本文研究并指出了进一步提高二维有限点集或线段集的凸壳生成算法效率的改进与优化方向。

2　二维凸壳问题与凸壳算法简介

自然,二维有限线段集,可看成只是二维有限点集的特例(即:二维点集中的若干"点对",恰好又分别构成各自线段),故二维有限点集或线段集的凸壳问题研究可归结为研究二维有限点集凸壳问题。

定义 1　设多边形 Q 的顶点是给定平面内的点 $Q_1(x_1,y_1),Q_2(x_2,y_2),\cdots,Q_n(x_n,y_n)$。如果任意线段 Q_iQ_j $(i\neq j,1\leqslant i\leqslant n,1\leqslant j\leqslant n,3\leqslant n<+\infty)$总不在多边形 Q 外,则称 Q 为凸多边形。

定义 2　设二维点集 $S=\{P_i(x_i,y_i)\mid 1\leqslant i\leqslant m,3\leqslant m<+\infty\}$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 是可覆盖 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。

不言而喻,如果凸多边形 Q 为二维点集 S 的凸壳,则必定 $3\leqslant n\leqslant m<+\infty$。

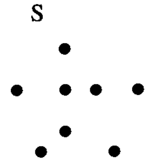
定义 3　如何寻求给定二维有限点集 $S=\{P_i(x_i,y_i)\mid 1\leqslant i\leqslant m,3\leqslant m<+\infty\}$的二维凸壳,称为二维凸壳问题。

通俗地讲,二维凸壳问题的几何原型,可简单而形象地说明如图 1 所示。

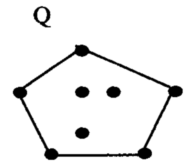
设 $1\leqslant i\leqslant m,3\leqslant m<+\infty$,对木板上的有限点集 S 中各点 $P_i(x_i,y_i)$处,分别各钉一个图钉,再用一条橡皮带从该点

*)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。

集外沿去围绕这些图钉。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。



A) 木板上各点处各钉一个图钉：二维点集 S



B) 外围缠绕各图钉所得橡皮带圈：二维凸壳 Q

图 1　二维凸壳问题几何原型的形象说明示意图

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

长期来，二维凸壳问题及凸壳生成算法（简称凸壳算法）颇受学者和业界关注，并已提出和应用了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[4~11]。限于篇幅，本文在此仅以较具代表性的卷包裹凸壳算法与格雷厄姆凸壳算法为例，阐明凸壳生成算法的改进与优化方向。

3　卷包裹凸壳算法评述

1970 年，D. Chand 和 S. Kapur 在"An algorithm for convex polytopes"[9]一文中，提出了卷包裹凸壳算法。

3.1　卷包裹凸壳算法思想简介

卷包裹凸壳算法，实际上就是图 1 橡皮带外绕法的具体计算机模拟实现。其算法思想可概括为：

第 0 步：取二维有限点集 S 的点 $P_1(x_1, y_1)$，使 $y_1=\min\{y_i \mid 1\leqslant i\leqslant m, 3\leqslant m<+\infty\}$；显然，点 P_1 就是凸壳 Q 的初始顶点 Q_1（即第 1 个顶点）；过点 Q_1 画一水平直线 L。

第 1 步：使 L 绕 Q_1 按逆时针方向旋转若干个转角 Δ，可碰到 S 中的第 2 个点 P_2；而点 P_2 就是凸壳 Q 的第 2 个顶点 Q_2，且线段 Q_1Q_2 就是凸壳 Q 的第 1 个边。

……

第 i 步：使 L 改绕顶点 Q_i 按逆时针方向旋转若干个转角 Δ，可碰到 S 中的第 $i+1$ 个点 P_{i+1}；而点 P_{i+1} 就是凸壳 Q 的第 $i+1$ 个顶点 Q_{i+1}，且线段 Q_iQ_{i+1} 就是凸壳 Q 的第 i 个边。其中，$2\leqslant i\leqslant n-1, 3\leqslant n\leqslant m<+\infty$。

……

第 $n-1$ 步：使 L 改绕顶点 Q_{n-1} 按逆时针方向旋转若干个转角 Δ，可碰到 S 中的第 n 个点 P_n；而点 P_n 就是凸壳 Q 的第 n 个顶点 Q_n，且线段 $Q_{n-1}Q_n$ 就是凸壳 Q 的第 $n-1$ 个边。

第 n 步：使 L 改绕顶点 Q_n 按逆时针方向旋转若干个转角 Δ 而转回顶点 Q_1，且线段 Q_nQ_1 就是凸壳 Q 的第 n 个边。

最后，由封闭折线 $Q_1Q_2, Q_2Q_3, Q_3Q_4, \cdots, Q_{n-1}Q_n, Q_nQ_1$ 构成的凸多边形，必定是所求二维有限点集 S 的凸壳 Q。

3.2　卷包裹凸壳算法主要弱点

应当指出：卷包裹凸壳算法，显然仅可无误差地确定其初始顶点 Q_1，而对其它顶点 $Q_i(i>1)$ 的确定都可能存在误差，即：其凸壳精度、算法效率，都取决于：它每次逆时针方向旋转时的固定转角 Δ 必须足够小；否则，往往会出现误差（详见下其缺点分析）。所以，它只是近似算法，并有下列两个主要弱点。

第一、卷包裹凸壳算法的转角 Δ，是制约该算法成效好坏、甚至成败的关键：若转角 Δ 设定过小，则有可能连续旋转多个转角 Δ，也碰不到一个欲寻找的凸壳顶点，故此时其转角扫描处理可谓"无用功较多，白耗时不少"；从而，使其凸壳精度较高但算法效率偏低。反之，该转角 Δ 若设定过大，则有可能才仅仅旋转了一个转角 Δ，就使一个甚至多个原本该找出的凸壳顶点被其"一转而过"——已被错误地漏掉，从而使得所找出的凸壳容易出现较大误差；故此时，其凸壳算法效率较高而精度偏低。

第二、它只能进行串行计算，而不利并行化。这是因为：(1)显然，只有凸壳的初始顶点 Q_1，才可以通过寻找二维有限点集的其 Y 轴座标值最小点而唯一确定；故仅有此第一个顶点 Q_1，才可保证自身全无误差。(2)当 $i\geqslant1$ 时，卷包裹凸壳算法每次所生成凸壳的下一顶点 Q_{i+1}，都要完全依赖于其前一顶点——当前顶点 Q_i。换言之，对当前刚找出的最新顶点 Q_i 来说，卷包裹凸壳算法最多只能找出 Q_i 的下一顶点 Q_{i+1}，而全然无法只用凸壳当前顶点 Q_i，就能找出该顶点 Q_i 后的其它各顶点 Q_{i+k}，这里 $k>1$。因此，这种直接前趋顶点依赖性充分说明：基于卷包裹凸壳算法，是不可能被改造为"能并发地对凸壳同时展开寻找其不同顶点的并行处理"。

4　格雷厄姆凸壳算法评述

1972 年，R. Graham 在"An Afficient Algorithm for Determining the Convex Hull of a Finite Planar set"[10]一文中，提出了格雷厄姆凸壳算法。

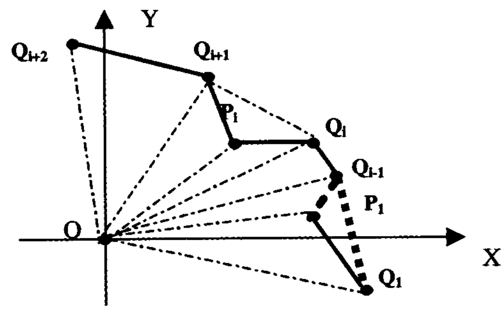


图 2　格雷厄姆算法关键（即格雷厄姆三角形）示意图

4.1　格雷厄姆凸壳算法思想简介

格雷厄姆凸壳算法（或称格雷厄姆搜索），是人们迄今仍使用较多的凸壳算法。其算法思想可概括为：

第 0 步：任取二维有限点集 S 的内点 P_0 作为坐标原点 O，如图 2 所示。

第 1 步：首先，求点集 S 中各点 P_i 相对于 OX 方向的转角$\angle P_iOX$（即：旋转 X 轴到点 P_i 时，与 OX 方向所成夹角，这

里 $0\leqslant\angle P_iOX\leqslant 2\pi$);然后,按其转角$\angle P_iOX$,升序排列点集 S 中其余各点,并仍记之为 P_i,$1\leqslant i\leqslant m$,$3\leqslant m<+\infty$。其中:$\angle P_1OX$ 为最小转角,并取该点 P_1 作为凸壳 Q 的初始顶点 Q_1。

……

第 i 步:删除格雷厄姆三角形$\triangle Q_iOQ_{i+1}$的所有内点并记之为点 P_i。即:如图 2 所示,如果某内点 P_i 不是凸壳顶点,则它必为位于原点 O 与它所在最邻近的两个凸壳顶点 Q_i、Q_{i+1}所构成格雷厄姆三角形 Q_iOQ_{i+1}的内点,故删除这些内点 P_i。其中,$1\leqslant i<n-1\leqslant m<+\infty$。

……

第 $n-1$ 步:删除格雷厄姆三角形 $Q_{n-1}OQ_n$ 的所有内点 P_{n-1}。

第 n 步:删除格雷厄姆三角形 Q_nOQ_1 的所有内点 P_n。

最后,由所得顶点序列 $Q_1,O_2,Q_3,\cdots,Q_{n-1},O_n,Q_1$ 顺次连接其相邻两顶点而得到的凸多边形 Q,必定是所求二维有限点集 S 的凸壳。

4.2 格雷厄姆凸壳算法主要弱点

显然,格雷厄姆凸壳算法也是近似算法,并且存在如下三个主要弱点。

第一、凸壳的初始顶点 Q_1 与其转角$\angle P_1OX$ 并无必然直接联系,故确定其初始顶点 Q_1 的"把 P_1 直接作为 Q_1"的处理方式,可能潜伏误判的误差甚至隐患。其所以如此的基本原因是:首先,其坐标原点 O 是任取自二维有限点集 S 的内点 P_0,故即使第一个内点 P_1 的转角$\angle P_1OX$ 为所有内点转角集合$\{\angle P_iOX \mid 1\leqslant i\leqslant m,3\leqslant m<+\infty\}$的最小者,也无法保证"二维有限点集 S 的第一个内点 P_1,就一定是其凸壳 Q 的第一个顶点 Q_1"。例如:图 2 中,显然有 $0<\angle P_1OX<\pi/2$,$0<\angle P_iOX<\angle XOQ_1<2\pi$,但此时(即位于圆点状虚线处的)点 P_1 仅为凸壳的内点,而绝不能错误地认为"该点 P_1 就是凸壳的第一个顶点,尽管点 P_1 的转角最小",因为此时"只有图中最下面的其转角最大的(即位于方块状虚线处的)点 Q_1,才真正是所求二维有限点集 S 的凸壳顶点"。

顺便指出:格雷厄姆凸壳算法可能产生近似性的唯一缺点——初始顶点近似性,可用卷包裹凸壳算法的初始顶点 Q_1 决定法来改进和消除,即"不再以二维有限点集的转角最小点 P_1 为其凸壳的初始顶点,而改为以二维有限点集的其 Y 轴座标值最小点 Q_1 作其凸壳的初始顶点"。

第二、格雷厄姆三角形内点判定的理论基础较差与处理效率较低,是限制该算法效率提高的瓶颈。一方面,格雷厄姆对内点 P_i 的判定基础,必须以格雷厄姆三角形$\triangle Q_iOQ_{i+1}$存在为前提;而格雷厄姆三角形$\triangle Q_iOQ_{i+1}$的确定,又只能基于当前顶点 Q_i,才能顺次求得其下一顶点 Q_{i+1},且 Q_{i+1} 的确定实际上只能在"能断定 Q_{i+1} 不是$\triangle Q_iOQ_{i+1}$的内点 P_i"之时。这就不能不使格雷厄姆凸壳算法存在天生缺陷。另一方面,由于对每一个格雷厄姆三角形$\triangle Q_iOQ_{i+1}$都要进行内点判定,而格雷厄姆三角形$\triangle Q_iOQ_{i+1}$中的可能内点又往往为数众多,且每一内点 P_i 的判定处理的效率通常难以提高,故必然使格雷厄姆凸壳算法的整体处理效率较低。

第三,格雷厄姆凸壳算法,只适于串行计算,而不利并行化。由于构造格雷厄姆三角形$\triangle Q_iOQ_{i+1}$必须以当前顶点 Q_i 为基础,这就使格雷厄姆凸壳算法也存在类似卷包裹凸壳算法的"直接前趋顶点依赖性"缺点,故它同样不适宜改造为并行化处理。

5 凸壳传统算法研究反思与同构化凸壳构造基本定理

20 世纪提出的凸壳问题,其算法研究始于 20 世纪 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。要改变这种窘况,人们必须率先正视下列凸壳自身客观存在的明显而简单的基本特性现象:

ⅰ.必存在"凸壳各边与其壳内点无关性"(可称顶点对内点的"凸壳内点无关性")的特性现象。即"凸壳 Q 的任一条边(简称凸边,即 Q_iQ_j,其中:$|\ i-j\ |=1$ 或 $n-1$,$1\leqslant i\leqslant n$,$1\leqslant j\leqslant n$,$3\leqslant n<+\infty$),仅唯一地取决于它自身的两个端点(即凸壳顶点 Q_i、Q_j),而完全与凸壳 Q 内部的任一内点 P_j(其中:$P_j\neq Q_k$,$1\leqslant j\leqslant m$,$1\leqslant k\leqslant n\leqslant m<+\infty$,$3\leqslant n$)全然无关"。

ⅱ.总存在"凸壳顶点与其它顶点无关性"(可称顶点对顶点的"凸壳顶点无关性")的特性现象。即"凸壳 Q 的任一顶点 Q_k(其中:$1\leqslant k\leqslant n<+\infty$,$3\leqslant n$),仅唯一地取决于它自身,而完全与除它以外的任何顶点 Q_i(其中:$i\neq k$,$1\leqslant i\leqslant n\leqslant m<+\infty$,$3\leqslant n$)全然无关"。

但事实上,深入分析与认真反思凸壳传统算法研究的优点、弱点及其原因,发现现行凸壳算法研究现状实在令人疑惑与颇感遗憾:迄今为止,现行二维点集或线段集凸壳算法(无论卷包裹算法、凸壳论格雷厄姆凸壳算法,还是其它现行凸壳算法)及其研究,都一直尚未能深刻认识、充分研究、高效利用"凸壳内点无关性"与"凸壳顶点无关性"这两大重要而基本特性。

依据凸壳的"内点无关性"与"顶点无关性"同构化现象,作者于 2005 年率先总结和提出了如下基本而重要的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 $S=\{P_i(x_i,y_i)\mid 1\leqslant i\leqslant m,3\leqslant m<+\infty\}$,凸多边形 Q 的顶点集为 $R=\{Q_i(x_i,y_i)\mid 1\leqslant i\leqslant n\leqslant m<+\infty,3\leqslant n,3\leqslant m\}$。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 $Q_i(x_i,y_i)\notin S-R$,$1\leqslant i\leqslant n\leqslant m<+\infty$,$3\leqslant n$,$3\leqslant m$。

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 $Q_k(x_k,y_k)\notin R-\{Q_k\}$,$1\leqslant k\leqslant n\leqslant m<+\infty$,$3\leqslant n$,$3\leqslant m$。

证明:

ⅰ.假设凸壳 Q 的某个顶点 $Q_k(x_k,y_k)\in S-R$,$1\leqslant k\leqslant n$;则必有顶点 $Q_k(x_k,y_k)\notin R$(这表明:该顶点 $Q_k(x_k,y_k)$应为凸壳 Q 的内点)。但 R 为凸壳 Q 的顶点集,故由已知条件可知,应有顶点 $Q_k(x_k,y_k)\in R$。这两者显然矛盾,故命题ⅰ成立。(这表明:凸壳 Q 的任何内点,必定不是凸壳 Q 的顶点)。

ⅱ.设 Q_k 为凸壳 Q 的任一顶点,则 Q_k 两侧至少有两个紧邻顶点 Q_{k-1} 与 Q_{k+1}(其中:$k\in R$;若 $k-1=0$,则取 $k-1$ 为 n;若 $k+1=n+1$,则取 $k+1$ 为 1)。显然,Q_{k-1}、Q_k、Q_{k+1}三点构成一个紧邻顶点三角形 $Q_{k-1}Q_kQ_{k+1}$,而顶点 Q_k 的位置是确定不变的,且只与自身坐标位置有关,而与其两个紧邻顶点 Q_k、Q_{k+1}的坐标位置无关。以此类推,可知:顶点 Q_k 的位置只与自身坐标位置有关,而与其两侧的所有相邻顶点 Q_i 与

(下转第 247 页)

上角是一个小地图，便于用户定位自己的当前位置及纵览整个博物馆。

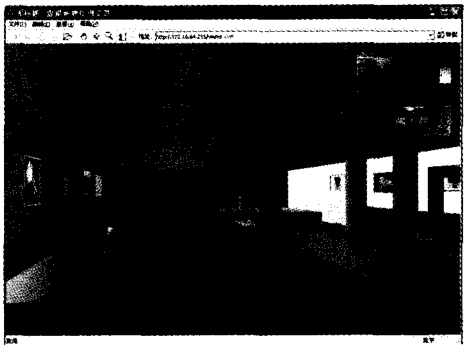


图 7　虚拟漫游界面

结论和进一步工作　本文概述了虚拟博物馆系统的设计和实现的基本过程和关键技术。系统采用 Visual C^{++}6.0 编程实现，并使用多种设计模式，大大降低了模块间的耦合程度，增强了稳定性和鲁棒性，具有良好的自适应能力。其中，一些关键技术如场景描述语言 XVM，为解决数字博物馆的虚拟展示问题提供了一条新的思路。目前，已完成的南京大学地球科学博物馆系统已经投入使用，获得了良好的效果。

下一步需要研究的课题包括：当场景中的 3D 模型较多时，如何根据视点的位置和方向以及网络负载等因素合理安排模型下载次序，并在 XVM 文档结构中提供相应的支持；以及需要在 XVM 中增加对物体行为的描述，加强虚拟博物馆地交互能力。

参 考 文 献

1　Addison A C. Emerging Trends in Virtual Heritage [J]. IEEE Multimedia, 2000, 7(2): 22～25

2　Kwon Y M, Kim I J, Ahn S C, et al. Networked 3D Virtual Museum System. [J]. Journal of System Simulation, 2003, 15(3): 301～305

3　Vince J. Virtual Reality System [M]. SIGGRAPH Series, ACM Press, Addison-Wesley Publishing Company, 1995

4　向辉，孟祥旭，杨承磊. 山东大学考古数字博物馆设计与实现 [J]. 系统仿真学报, 2003, 15(3): 319～321

5　Iwazaki K, Yasuda T, Yokoi Shigeki, et al. The Museum Network and On demand Systems for School Education Based on XML. In: International Conference on Computers in Education, 2002. 941～942

6　Lin M C, Gottschalk S. Collision detection between geometric models: a survey [A]. In: Proceedings of IMA Conference on Mathematics of Surfaces, 37～56

7　Martin R C. Agile Software Development, Principles, Patterns, and Practices. Reading, MA: Person Education, 2003

8　Gamma E, Helm R, Johnson R, et al. Design Patterns: Elements of Reusable Object-Oriented Software. Reading, MA: Addison-Wesley, 1995

9　Schmidt D C. Reactor: An Object Behavioral Pattern for Concurrent Event Demultiplexing and Handler Dispatching. In: Pattern Languages of Program Design (J. O. Coplien and D. C. Schmidt, eds.), Reading, MA: Addison-Wesley, 1995. 529～545

10　Jain P, Schmidt D C. Service Configurator: A Pattern for Dynamic Configuration of Services. In: Proc. of the 3rd Conference on Object Oriented Technologies and Systems, USENIX, June 1997

（上接第 218 页）

Q_j（其中：$i \neq k, j \neq k, i \neq j, k \in R, i \in R, j \in R$）的坐标位置无关，故凸壳 Q 的任一顶点 Q_k 独立于其它顶点。

因此，该基本定理成立。

结论　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行并行化改造与创新。无疑，这必定是今后“突破目前凸壳算法研究停滞不前窘况，进一步提高凸壳算法（包括串行、并行）效率”的主要捷径。

实际上，作者利用同构化凸壳构造基本定理和本文结论，于 2005 年已研究出诸如“基于最大基线倾角智能逼近的凸壳新算法”等凸壳新算法。

参 考 文 献

1　张立华，徐文立. 基于凸壳的透视变换下的点模式匹配方法[J]. 自动化学报，2002(2)

2　(美) Gonzalez R C, Richard E. Word. 数字图象处理[M]. 阮秋琦，等译. 北京：电子工业出版社，2003

3　任群，田捷. 基于前景轮廓线搜索的指纹图象分割算法[M]. 北京：中科院自动化所复杂科学与智能系统实验室

4　徐常青，等. 计算机图形学[M]. 机械工业出版社，2004

5　周培德. 计算几何 算法分析与设计[M]. 清华大学出版社，2000

6　陈国良. 并行计算 结构·算法·编程[M]. 高等教育出版社，2002

7　陈国良. 并行算法的设计与分析[M]. 高等教育出版社，2002

8　Aloupis G. A History of Linear-time Convex Hull Algorithms for Simple Polygons. http://en.wikilib.com/wiki/Talk:Convex_hull

9　Chand D, Kapur S. An algorithm for convex polytopes. J. ACM, 1970, 17: 78～86

10　Graham R. An efficient algorithm for determining the convex hull of a finite planar point set. Info. Proc. Letters, 1972, 1: 132～133

11　Barber C, Dobkin D, Huhdanpaa H. The Quickhull algoritm for convex hulls. ACM Trans. on Mathematical Software, 1997, 22: 469～483

（上接第 243 页）

我们还进一步结合细节点提取和匹配的结果来评估增强算法的性能。实验结果表明：所提出的算法能有效减少细节点提取的错误和提高指纹识别的准确性。

参 考 文 献

1　O'Gorman L, Nickerson V. An approach to fingerprint filter design. Pattern Recognition, 1989, 22(1): 29～38

2　Sherlock B G, Monro D M. Millard K. Fingerprint enhancement by directional Fourier filtering. In: IEE Proc. Vision Image Signal Process, 1994, 141(2): 87～94

3　Hsieh C T, Lai E, Wang Y C. An effective algorithm for fingerprint image enhancement based on wavelet transform. Pattern Recognition, 2003, 36(2): 303～312

4　Hong L, Wan Y, Jain A. Fingerprint image enhancement: algorithm and performance evaluation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, 20(8): 777～789

5　Chikkerur S, Govindaraju V, Cartwright A N. Fingerprint image enhancement using SIFT analysis. In: Proceedings of 3rd International Conference on Advances in Pattern Recognition, Bath, UK, 2005. 20～29

6　Field D J. Relations between the statistics of natural images and the response properties of cortical cells. Journal of the Optical Society of America A, 1987, 4(12): 2379～2394

7　Nagaty K A. On learning to estimate the block directional image of a fingerprint using a hierarchical neural network. Neural Networks, 2003, 16(1): 133～144

8　Zhou J, Gu J. A model-based method for the computation of fingerprints'orientation field. IEEE Transactions on Image Processing, 2004, 13(6): 821～835

9　Kovacs-Vajna Z M, Rovatti R, Frazzoni M. Fingerprint ridge distance computation methodologies. Pattern Recognition, 2000, 33(1): 69～80

10　Jea T Y, Govindaraju V. A minutia-based partial fingerprint recognition system. Pattern Recognition, 2005, 38(10): 1672～1684