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

周启海[1,2] 黄 涛[1,2] 吴红玉[2]
(西南财经大学信息技术应用研究所 成都 610074)[1] (西南财经大学经济信息工程学院 成都 610074)[2]

摘 要 本文针对现行凸壳算法(诸如:串行类的卷包裹凸壳算法、格雷厄姆凸壳算法等,并行类的折半分治凸壳算法、快速凸壳算法等)效率不高的缺点,根据同构化凸壳构造基本定理,利用工作站机群优点,提出了效率更高的双群(即:其机群分为2个子机群)、双域(即:其数据分布域分为2个子分布域)、四向(即:其每个子分布域内凸壳顶点的寻找方向均各自为顺时针、逆时针2个寻找方向)水平倾角最小化圈绕的凸壳并行新算法。

关键词 同构化,机群,凸壳,并行算法,双群,双域,四向

A New Parallel Algorithm for Finding Convex Hull Based on Minimum Lever Pitch Coiling with 2-Clusters, 2-Domains and 4-Directions

ZHOU Qi-Hai[1,2] HUANG Tao[1,2] WU Hong-Yu[2]
(Research Institute of Information Technology Application, Southwestern University of Finance and Economics, Chengdu 610074)[1]
(School of Economic Information Engineering, Southwestern University of Finance and Economics, Chengdu 610074)[2]

Abstract In this paper, comment on the lower efficiency shortcomings of representative both series algorithms for finding convex hull (for example: Gift wrapping convex hull algorithm, Graham scan convex hull algorithm, and so on) and parallel algorithms for finding convex hull(for example: Half-dividing convex hull algorithm, Rapid convex hull algorithm, and so on), and according to the isomorphic fundamental theorem of the convex hull construction and using the advantages of COW (Cluster of workstation), a more efficient new parallel algorithm to find a convex hull is given. The general characters of the new parallel algorithm are: 1) its COW is combined with two sub-clusters; 2) its domain is divided into two sub-domains; 3) its seeking directions of coiling with a minimum lever pitch in every sub-domain are along with two ways (i. e. clockwise direction, and anti clockwise direction) separately.

Keywords Isomorphic, COW, Convex hull, Parallel algorithm, 2-Clusters, 2-Domains, 4-Directions

20世纪70年代以来,二维凸壳(Convex hull)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,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,已提出不少串行凸壳算法(诸如:卷包裹凸壳算法、格雷厄姆凸壳算法等),也提出一些并行凸壳算法(例如:折半分治凸壳算法、快速凸壳算法等)。2005年,作者开始潜心研究凸壳算法,现已研究、改进和提出了一些串行凸壳新算法[1~7]。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,基于周启海教授提出的同构化凸壳构造基本定理[3],本文提出了效率更高的基于工作站机群的双群双域四向水平倾角最小化圈绕的凸壳并行新算法。

1 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 $Q_1(x_1, y_1), Q_2(x_2, y_2), \cdots, Q_n(x_n, y_n)$。如果线段 $Q_iQ_j(i\neq j, 1\leqslant i\leqslant n, 1\leqslant j\leqslant n)$ 总不在多边形 Q 外,则称 Q 为凸多边形。

定义2 设二维点集 $S=\{P_i(x_i, y_i) \mid 1\leqslant i\leqslant m,\ 3\leqslant m<\infty\}$ 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。如何寻求给定二维点集 $S=\{P_i(x_i, y_i) \mid 1\leqslant i\leqslant m,\ 3\leqslant m<\infty\}$ 的二维凸壳,称为二维凸壳问题。凡能构造性生成给定二维点集 $S=\{P_i(x_i, y_i) \mid 1\leqslant i\leqslant m,\ 3\leqslant m<\infty\}$ 的二维凸壳的算法,统称二维凸壳生成算法,简称二维凸壳算法(或凸壳算法)。

2 并行算法概要

并行程序设计方法不同于串行程序设计方法,其主要差别在于:串行程序设计方法把事务的变化发展看成是单线程的,任何两种事务之间必然存在因果关系,从而把一系列相关事务看成是不可分割的整体。在对事务的认识上,结构程序设计特别是对象程序设计(即面向对象程序设计),应该说取得了突破性的进展(即:它们已把一个复杂的事务分解成多个简单事务,甚至把一个系统看成是多个相互联系的实体所构成);但两者都仅着眼于事务的静态结构关系与表层行为模式,尚未实现从动态结构关系与深层行为模式来认识和分解事务。因而,从行为方式的本质来看,事务仍然被认为是一致的,或者是先后相继的,相互之间不存在并发性工作方式和干扰现象,更不存在可同时发生的相互作用的并发行为。并行程序设计方法的一个最基本的观点,就是把一个事务的行为看成是多个(串行或并行的)子事务互相作用的结果。这是程

周启海 教授,博(硕)士生导师,主要研究方向:计算几何,算法研究与应用,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。

序设计观念的根本转变,它主导下的并行程序设计方法,其核心内容就是事务的并行划分与算法映射,其理论基础是并行计算模型。并行计算模型决定了并行语义,并行语义决定了可并行执行的准则,从而决定了并行划分原则与并行算法设计。鉴于分布并行算法的广泛性和主流性,本文所论凸壳并行算法采取分布并行计算实现方式。

定义 3 并行算法(Parallel Algorithm),是一些同时执行、相互作用、彼此协调、目标统一、协同一致地完成给定问题求解的多个进程的集合。

定义 4 采用同步方式(指算法的诸进程的执行必须相互等待)、异步方式(指算法的诸进程的执行不必相互等待)、分布方式(指由通信链路连接的多个场点 Size 或节点 node 来协同完成问题求解)进行并行处理的并行算法,分别称为同步并行算法(Synchronized Algorithm)、异步并行算法(Asynchronized Algorithm)、分布并行算法(Distributed Algorithm)。

3 工作站机群 COW 简介

工作站机群简称机群(COW,即 Cluster Of Workstation),是利用高速通用网络将一组高性能工作站(或高档 PC 机),按某种拓扑结构连接起来,并在并行程序设计以及可视化人机交互集成开发环境支持下,进行统一调度、协调处理、高效实现高性能计算的并行处理系统,COW 的每个结点上都有完整的操作系统。从系统结构和结点间通信方式来看,COW 属于分布式存储的 MIMD 并行计算机结构,它主要利用消息传递方式实现各主机之间的通信,由建立在一般操作系统之上的并行编程环境完成系统的资源管理及相互协作,同时也屏蔽工作站及网络的异构性。因此,对程序员和用户来说,机群系统是一个整体的并行处理系统。机群系统中的主机和网络既可以是同构的,也可以是异构的。其系统结构如图 1 所示。

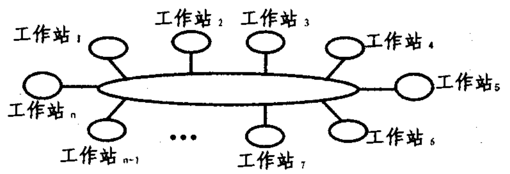


图 1 机群 COW 的一般系统结构示意图

机群系统中,各工作站(含其下属的每台处理机)都有自己的存储器和 I/O 设备,其基本架构如图 2 所示。机群系统,具有如下重要基本特点:

1)环境适应性强,系统开发周期短。开发的重点在通信和并行编程环境上;它既不用重新研制计算节点,又不用重新设计操作系统和编译系统,这就节省了大量的研制时间。

2)系统鲁棒性好,用户投资风险小。机群系统不仅是一个并行处理系统,它的每个节点同时也是一台独立的工作站,即使整个系统对某些应用问题的并行效率不高,它的结点仍然可以作为单个工作站使用。

3)系统性价比好,系统开发成本低。传统巨型机或 MPP 因其生产的小批量性,故其成本昂贵,使其售价甚高(动辄往往数百万到上千万美元)。而工作站(或高档 PC 机)属于大批量生产,故其成本与售价自然低廉得多。

4)系统利用率甚高,系统资源颇节省。它可以充分利用现有设备,故单从使用效率上看,机群系统的资源利用率也比单机系统要高得多。

5)系统易延展性好,系统可扩充性强。从规模上说,机群系统大多使用通用网络,系统扩展容易;从性能上说,对大多数中、粗粒度的并行应用都有较高的效率。

6)资源重用度高,用户编程纠集简便。

机群系统中,程序的并行化通常只是在原有串行环境中插入相应的通信原语,而对串行程序资源库也往往只需做并不很多的修改。

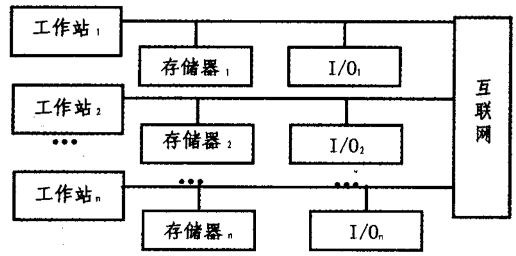


图 2 机群 COW 中各工作站的基本架构示意图

因此,本文采用机群技术来研究和实现凸壳算法的并行化。

4 双群双域四向的并行凸壳新算法

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

定义 6 二维点集 $S=\{P_i(x_i,y_i)\mid 1\leqslant i\leqslant m\geqslant 3\}$ 中,其 Y 轴坐标值最大、最小的两个最外点分别记为 $P_{(1)}(x_1,\ y_1=\min\{y_i(1\leqslant i\leqslant m\geqslant 3)\})$,$P_{(2)}(x_2,\ y_2=\max\{y_i(1\leqslant i\leqslant m\geqslant 3)\})$;这两个最外点 $P_{(1)}$、$P_{(2)}$,合称 S 的凸壳 Q 的初始极点,并记为 $Q_{上0}$、$Q_{下0}$(意即:$Q_{上0}$ 同为子分布域 $S_{右}$、$S_{左}$ 的右上端的初始极点;同理,$Q_{下0}$ 亦然)。连接 $Q_{上0}$、$Q_{下0}$ 这两个点所得的直线,称为点集 S 分布域的分界基线(简称 S 基线);S 基线将二维点集 S 分布域分为两个小区域 $S_{左}$、$S_{右}$,称这两个小区域为二维点集 S 的子分布域(其示例如图 3)。

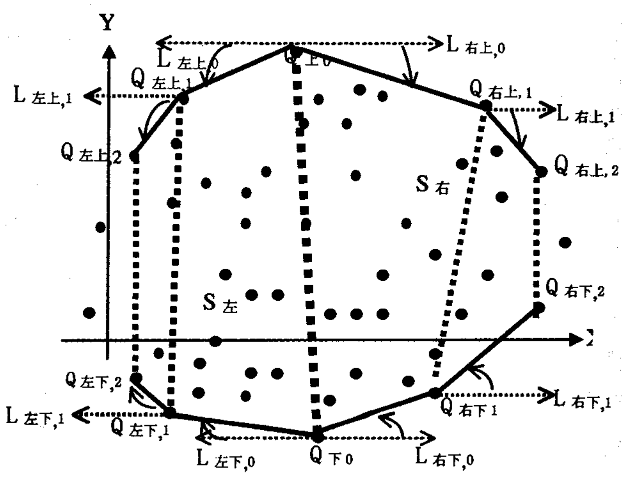


图 3 凸壳的初始极点、子分布域示意图

定义 7 不失一般性,二维点集 $S=\{P_i(x_i,y_i)\mid 1\leqslant i\leqslant m\geqslant 3\}$ 的凸壳 Q 的初始顶点 $Q_{上0}$、$Q_{下0}$,还可分别记为 $Q_{右上,0}$、$Q_{右下,0}$ 或者 $Q_{左上,0}$、$Q_{左下,0}$。在 S 的子分布域 $S_{右}$ 中,分别过 $S_{右}$ 的子凸壳 $Q_{右}$ 的当前新顶点 $Q_{右下,j}$、$Q_{右上,j}(0\leqslant j\leqslant m_{右})$,所作平行 X 轴正方向的同向顶点射线 $Q_{右下,j}L_{右下,j}$ 和 $Q_{右上,j}L$

右$_{上,j}$，称为顶点$Q_{右下,j}$、$Q_{右上,j}$的正向射线。正向射线$Q_{右下,j}L_{右下,j}$、$Q_{右上,j}L_{右上,j}$分别按逆时针方向（简称A向）、顺时针方向（简称B向）绕行到点$P_j(x_j,y_j)\in S_{右}$所成夹角$\angle P_iQ_{右下,j}L_{右下,j}$、$\angle P_iQ_{右上,j}L_{右上,j}$，分别称为点$P_j(x_j,y_j)$对正向射线$Q_{右下,j}L_{右下,j}$的$A$向水平倾角、正向射线$Q_{右上,j}L_{右上,j}$的$B$向水平倾角。同理，在$S$的子分布域$S_{左}$中，可定义点$P_r(x_r,y_r)\in S_{左}$对正向射线$Q_{左上,k}L_{左上,k}(0\leqslant k\leqslant m_{左})$的$A$向水平倾角$\angle P_rQ_{右上,k}L_{右上,k}$、正向射线$Q_{左下,k}L_{左下,k}(0\leqslant k\leqslant m_{左})$的$B$向水平倾角$\angle P_rQ_{左下,k}L_{左下,k}$。

作者，在文[1,2]中指明了凸壳算法改进与优化的同构化方向，在文[4～6]中阐明了对文[3]的改进（其中：文[5]又是对文[4]的改进）。在文[6]的基础上，本文进一步提出基于机群的双群（即：其机群分为2个子机群）、双域（即：其数据分布域分为2个子分布域）、四向（即：其每个分布域内凸壳顶点的寻找方向均各自为顺时针、逆时针2个寻找方向）并行凸壳新算法。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)“寻找分布域S的Y轴坐标值最大、最小的最外点”的双群并行处理：

1)标记构成所论机群COW的两个子机群为$COW_{右}$、$COW_{左}$；分别标记子机群$COW_{右}$、$COW_{左}$下属各处理机的总数为$n_{右}$、$n_{左}$；分别标记子机群$COW_{右}$、$COW_{左}$下属各处理机为$P_{右j}(1\leqslant j\leqslant n_{右})$、$P_{左k}(1\leqslant k\leqslant n_{左})$；设定初始分布域$S=\{P_i(x_i,y_i)\mid 1\leqslant i\leqslant m\geqslant 3\}$中各点的$X$轴坐标取值的最大、最小可能值分别为$x_{max}$、$x_{min}$。

2)并行地使子机群$COW_{右}$下属各处理机$P_{右j}(1\leqslant j\leqslant n_{右})$，对“把初始分布域$S$按初始带宽$W_{S宽}(=(x_{\max}-x_{\min})/(n_{右}+n_{左}))$作带状划分所得的”各自初始子分布域$S_{右初j}(1\leqslant j\leqslant n_{右})$内各点，如果初始子分布域$S_{右初j}$非空，则找出$S_{右初j}$内各点的$Y$轴坐标值最大、最小的两个最外点；再从所得各初始子分布域$S_{右初j}(1\leqslant j\leqslant n_{右})$的全部最外点中，找出各$S_{右初j}$的并集构成的右初始分布域$S_{右初}$的$Y$轴坐标值最大、最小的2个最外点。

3)并行地使子机群$COW_{左}$下属各处理机$P_{左k}(1\leqslant k\leqslant n_{左})$，对“把初始分布域$S$按初始带宽$W_{S宽}$作带状划分所得的”各自初始子分布域$S_{左初k}(n_{右}+1\leqslant k\leqslant n_{左}+n_{右})$内各点，如果初始子分布域$S_{左初k}$非空，则找出$S_{左初k}$内各点的$Y$轴坐标值最大、最小的两个最外点；再在所得各初始子分布域$S_{左初k}(n_{右}+1\leqslant k\leqslant n_{左}+n_{右})$的全部最外点中，找出各$S_{左初k}$的并集构成的左初始分布域$S_{左初}$的$Y$轴坐标值最大、最小的2个最外点。

4)使子机群$COW_{右}$、$COW_{左}$联合在右、左初始分布域$S_{右初}$、$S_{左初}$的Y轴坐标值最大、最小的4个最外点中，并行地找出初始分布域S的Y轴坐标值最大、最小的2个最外点，并分别记为$P_{(1)}(x_1,y_1=\max\{y_i(1\leqslant i\leqslant m\geqslant 3)\})$，$P_{(2)}(x_2,y_2=\min\{y_i(1\leqslant i\leqslant m\geqslant 3)\})$；该最外点$P_{(1)}$、$P_{(2)}$即为$S$的凸壳$Q$的初始极点（或顶点），并记为$Q_{上0}$、$Q_{下0}$。

(2)“构造分布域S的两个（即左、右）子分布域”的双群并行处理：

1)并行地分别使子机群$COW_{右}$、$COW_{左}$连接初始顶点$Q_{上0}$、$Q_{下0}$，生成初始分布域S的分界基线，以分划出供后续并行处理的二维点集S分布域的两个子分布域$S_{右}$、$S_{左}$。

2)并行地分别使子机群$COW_{右}$、$COW_{左}$，各自只挑选和保留子分布域$S_{右}$、$S_{左}$内各点数据；分别求出（“分别用处理机的总数$n_{右}$、$n_{左}$，对子分布域$S_{右}$、$S_{左}$中全部点各自进行按带状划分”所需的）子域$S_{右j}(1\leqslant j\leqslant n_{右})$、$S_{左j}(1\leqslant j\leqslant n_{左})$的带宽$W_{右宽}=(\max\{x_i\mid x_i\in S_{右}\}-\min\{x_i\mid x_i\in S_{右}\})/n_{右}$，$W_{左宽}=(\max\{x_i\mid x_i\in S_{左}\}-\min\{x_i\mid x_i\in S_{左}\})/n_{左}$。

第1步：“子机群$COW_{右}$、$COW_{左}$分别在子分布域$S_{右}$、$S_{左}$中生成子凸壳$Q_{右}$、$Q_{左}$各顶点”的双群、双域、四向水平倾角最小化并行圈绕处理。

第1-1步 子机群$COW_{右}$在子分布域$S_{右}$中，进行双向水平倾角最小化圈绕寻找子凸壳$Q_{右}$各顶点[5]的并行处理。

第1-1-1步 “子机群$COW_{右}$双向圈绕寻找子凸壳$Q_{右}$下一对新顶点”的并行处理：

第1-1-1-1步 初始子分布域$S_{右}$仍标记为当前子分布域$S_{右}$；标记初始顶点$Q_{上0}$、$Q_{下0}$为当前新顶点对$Q_{右上,0}$、$Q_{右下,0}$；置（当前新顶点对的序号计数器）r初值为0。

第1-1-1-2步 并行地使子机群$COW_{右}$下属各处理机$P_{右j}(1\leqslant j\leqslant n_{右})$，用带宽$W_{右宽}$把当前子分布域$S$带状划分为$n_{右}$个子处理域$S_{右j}(1\leqslant j\leqslant n_{右})$。

第1-1-1-3步 并行地使子机群$COW_{右}$下属各处理机$P_{右j}$各在自己的子分布域$S_{右j}(1\leqslant j\leqslant n_{右})$的全部点中，均以$Q_{右上,r}$、$Q_{右下,r}$为当前新顶点，求出其双向圈绕水平倾角最小化的$A$向、$B$向最小点$P_{右jA}$、$P_{右jB}$；在所得各组$A$向、$B$向最小点$P_{右jA}$、$P_{右jB}(1\leqslant j\leqslant n_{右})$中，找出当前子分布域$S_{右}$的$A$向、$B$向最小点$P_{右A}$、$P_{右B}$；置$r$为$r+1$，并使$P_{右A}$、$P_{右B}$作为子凸壳$Q_{右}$的下一对新顶点$Q_{右上,r}$、$Q_{右下,r}$。

第1-1-2步 删除由“所得凸壳当前次新顶点对$Q_{右上,r-1}$、$Q_{右下,r-1}$、最新顶点对$Q_{右上,r}$、$Q_{右下,r}$（$Q_{右上,r}$、$Q_{右下,r}$可能合而为一）所构成子凸壳$Q_{右上,r-1}Q_{右下,r-1}Q_{右下,r}Q_{右上,r}$的内点”的分布域极小化并行处理：

第1-1-2-1步 如果最新顶点对$Q_{右上,r}$、$Q_{右下,r}$不合而为同一点，

则：并行地使子机群COW右下属各处理机$P_{右j}(1\leqslant j\leqslant n_{右})$，用带宽$W_{右宽}=(\max\{x_i\mid x_i$为子凸壳$Q_{右上,r-1}Q_{右下,r-1}Q_{右下,r}Q_{右上,r}$内的点$\}-\min\{x_i\mid x_i$为子凸壳$Q_{右上,r-1}Q_{右下,r-1}Q_{右下,r}Q_{右上,r}$内的点$\})/n_{右})$，把子凸壳$Q_{右上,r-1}Q_{右下,r-1}Q_{右下,r}Q_{右上,r}$所在区域$QQ_{右}$带状划分为$n_{右}$个等宽子区域$QQ_{右j}(1\leqslant j\leqslant n_{右})$；

否则：转而执行第1-1-3步。

第1-1-2-2步 并行地使子机群$COW_{右}$下属各处理机$P_{右j}(1\leqslant j\leqslant n_{右})$，各自删除在自己的子区域$QQ_{右j}(1\leqslant j\leqslant n_{右})$中的全部点。

第1-1-2-3步 把删除$QQ_{右}$中全部点后的原子分布域$S_{右}$，仍标记为当前子分布域$S_{右}$；若当前子分布域$S_{右}$非空（即还有未找出的凸壳顶点），则回到第1-1-1-2步，否则转而执行第1-1-3步。

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

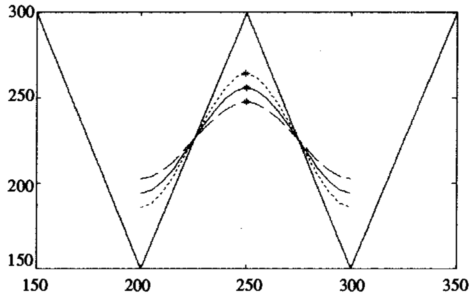
第1-2步 与上述第1-1步的处理方法同理，子机群$COW_{左}$在子分布域$S_{左}$中，进行双向水平倾角最小化圈绕寻找子凸壳$Q_{左}$各顶点的并行化处理。

第1-2-1步 “子机群$COW_{右}$双向圈绕寻找子凸壳$Q_{左}$下一对新顶点”的并行处理：

第1-1-2步 删除由“所得凸壳当前次新顶点对$Q_{左上,r-1}$、

（下转第241页）

同样可以推导，随着形状参数 λ 的增大，$k(k\geqslant3)$ 阶带形状参数的指数均匀 B 样条曲线更加逼近控制多边形顶点。由于证明过程比较繁琐，我们将另文说明。图 5 给出了形状参数 λ 取不同值时，曲线与控制点的逼近情况。

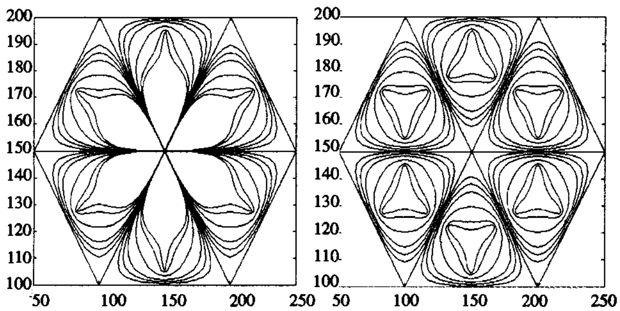


(虚线 $\lambda=-2$;实线 $\lambda=0$;点线 $\lambda=2$)

(虚线 $\lambda=-2$；实线 $\lambda=0$；点线 $\lambda=2$)

图 5　三阶带形状参数的指数均匀 B 样条曲线逼近

5　带形状参数的指数均匀 B 样条曲线的图形实例



(a) 花瓣　　(b) 花纹

图 6　四阶带形状参数的指数均匀 B 样条曲线造型图

曲线设计中的一个基本的内容是开曲线和闭曲线的构造，了解开曲线的端点行为和闭曲线的构造具有重要意义。对于控制点列 $P_i(i=1,2,\cdots,n)$，其中 $P_0=P_n$：若要构造开曲线，只须增加两个控制点 $P_0=2P_1-P_2$，$P_{n+1}=2P_n-P_{n-1}$，即可构造两端插值于 P_1、P_n，且在 t_0 和 t_n 处分别以 P_1-P_2、P_n-P_{n-1} 为切向量的开曲线。若要构造闭曲线，只须增加两个控制点 $P_{n+1}=P_2$，$P_{n+2}=P_3$。图 6 给出了 λ 取不同值时的开曲线和闭曲线。图 6(a)是 $\lambda=-2,-1,0,2,4,6$ 时，按控制点列生成的花瓣开曲线图形；图 6(b)是 $\lambda=-4,-2,-1,0,2,4,6,8$ 时，按控制点列生成的花纹闭曲线图形。不论是开曲线还是闭曲线，调整 λ 的值可以得到不同的曲线。当 λ 增大时，曲线逐渐远离控制多边形，当 λ 减小时候，曲线逐渐逼近控制多边形。

结论　本文构造了 $k(k\geqslant3)$ 阶带形状参数的指数均匀 B 样条样条基，并因此定义了带形状参数的指数均匀 B 样条曲线和曲面。该样条基具有与均匀 B 样条基相同的性质、结构，保持了均匀 B 样条曲线的一些实用的几何性质，并且该样条基带有一个形状参数，通过对形状参数取值的改变，可以调整曲线接近其控制多边形的程度。同时，该样条曲线可以构造张量积曲面，通过选取形状参数的值，也可以得到不同程度地接近其控制多面体的曲面。指数多项式均匀 B 样条定义线性空间 Ω_k 上，又由于 Ω_k 对于积分运算是封闭的，即对任意的 $N(t)\in\Omega_k$，有 $\int N(t)\mathrm{d}t\in\Omega_k$，为相关的积分运算带来方便。以上分析说明，指数多项式均匀 B 样条模型是自由曲线曲面造型的一个有力工具，具有良好的应用价值。

参 考 文 献

1　Zhang J W. C-curves: An extension of cubic curves. Com-puter-aided Geometric Design, 1996, 13(3): 199 ~217
2　Zhang J W. Two different forms of C-B-Splines. Computer-aided Geometric Design, 1997, 14(1): 31~41
3　Mainar E, Pena J M, Sanchez-Reyes J. Shape preser- ving alternatives to the rational Bezier model. Computer-aided Geo metric Design, 2001, 18(1): 37~60
4　Mainar E, Pena J M. A basis of C-Bezier like curves. Computer-aided Geometric Design, 2002, 19(4): 291~295
5　Han Xu-Li. Quadratic trigonometric polynomial curves with a shape parameter. Computer-aided Geometric Design, 2002, 19 (7): 503~512
6　Chen Qinyu, Wang Guozhao. A class of Bezier like curv- es. Computer-aided Geometric Design, 2003, 20(1): 29~39
7　Wang Guozhao, Chen Qinyu, Zhou Minghua. NUAT B- spline curves. Computer-aided Geometric Design, 2004, 21(2): 193~205

(上接第 234 页)

$Q_{左下,r-1}$、最新顶点对 $Q_{左下,r}$、$Q_{左上,r}$($Q_{左下,r}$、$Q_{左上,r}$ 可能合而为一)所构成子凸壳 $Q_{左上,r-1}Q_{左下,r-1}Q_{左下,r}Q_{左上,r}$ 的内点”的分布域极小化并行处理：

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

(注：因与上述第 1-1 步及其第 1-1-1 步、第 1-1-2 步、第 1-1-3 步的处理原理、方法、步骤、操作等极其类同，故除第 1-1 步中的“右、左、A 向、B 向”需对应改为第 1-2 步中的“左、右、B 向、A 向”外，第 1-2 步及其第 1-2-1 步、第 1-2-2 步、第 1-2-3 步的其它处理与操作均可从略。)

第 2 步　“将子机群 $COW_{右}$、$COW_{左}$ 所得子凸壳 $Q_{右}$、$Q_{左}$ 各顶点”组合为所求凸壳 Q 的全部顶点处理，即：顺序把上述已得子凸壳 $Q_{右}$、$Q_{左}$ 各顶点，依次两两连接而最后所得凸多边形 Q，必定是所求二维点集 S 的凸壳 Q。

结论　本文提出基于机群的双群双域四向凸壳并行新算法，不仅在时间、空间复杂度与效率上，均优于现行的卷包裹凸壳算法、格雷厄姆凸壳算法、折半分治凸壳算法等传统凸壳串行算法与并行算法；而且便于推广到基于机群的 m 群、n 域、p 向(其中：$m>2,n>2,\ p>2$)的凸壳并行新算法研究(对此，作者拟另文阐述)。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(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, 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(8)
5　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(9)
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(12)
7　周启海, 黄涛, 吴红玉. 双域双向水平倾角最小化圈绕凸壳新算法[C]. 见：中国几何设计与计算新进展(第三届中国几何设计与计算大会议论文集, 2007. 187~192