

# 海量点云数据轮廓特征线的快速生成算法

程效军<sup>1,2</sup>, 贾东峰<sup>1</sup>, 刘燕萍<sup>3</sup>

(1. 同济大学 测量与国土信息工程系, 上海 200092; 2. 同济大学 现代工程测量国家测绘地理信息局重点实验室, 上海 200092; 3. 同济大学 浙江学院 土木工程系, 浙江 嘉兴 314000)

**摘要:** 从海量点云数据中快速生成轮廓特征线, 是实现基于特征的模型重建的关键. 提出了一种基于切片的轮廓特征线快速生成算法. 该算法首先对点云数据进行切片, 将数字图像的方法应用到基于切片的特征点提取中, 通过设置数字栅格平面的边长快速地提取特征点, 并根据提出的双向索引连通法快速构造特征线, 最终实现了点云数据的曲线模型. 实例证明: 本算法可以快速、准确地生成海量点云数据的轮廓特征线.

**关键词:** 海量点云; 轮廓特征线; 切片; 数字图像; 曲线模型

**中图分类号:** P237

**文献标识码:** A

## A Fast Contour Generation Algorithm of Massive Point Cloud Data

CHENG Xiaojun<sup>1,2</sup>, JIA Dongfeng<sup>1</sup>, LIU Yanping<sup>3</sup>

(1. Department of Survey and Geo-informatics, Tongji University, Shanghai 200092, China; 2. Key Laboratory of Advanced Engineering Surveying of National Administration of Surveying, Mapping and Geoinformation, Shanghai 200092, China; 3. Department of Civil Engineering, Tongji University Zhejiang College, Jiaxing 314000, China)

**Abstract:** Rapid generation of contour from massive point cloud data is the crucial step for feature-based model reconstruction. This paper proposes a fast contour generation algorithm of massive point cloud data based on slicing. To extract the featured points (FPs) from slice, a layer-based point cloud model is directly generated from the point cloud data and serves as the input of a digital image based method. Meanwhile, the size of the grid of digital plane is discussed for fast getting FPs. Based on the FPs of each slice, bidirectional index connecting method (BICM) is put forward to quickly construct layer contour. Finally, the curve model is generated. Several case studies are presented to illustrate the

efficiency and accuracy of this approach.

**Key words:** massive point cloud; contour; slice; digital image; curve model

随着点云数据规模的不断增大, 如何有效地结合逆向工程(RE)和快速成型技术(RP), 从海量的点云数据中获取平面轮廓特征点, 并进行轮廓线的重建, 是实现基于特征的模型重建的关键. Lee 等<sup>[1]</sup>根据点的提取率对点云进行分层, 提出角偏差法来获取每层数据的曲率突变点, 并通过同伦算法生成点云的轮廓线模型. 由于曲率突变点的获取容易受到噪声的影响, 所以该方法抗噪能力差. Wu 等<sup>[2]</sup>提出了基于 RE 和 RP 的迭代算法, 通过定义形状因子, 计算每层数据的形状误差, 以此来调节点云分层的厚度, 实现分层划分的自适应, 但是其轮廓特征线的生成效率较低. Liu 等<sup>[3]</sup>提出了一种基于中间点的曲线模型法(IPCM)来集成 RE 和 RP. 该方法效率较高, 但未解决点云切片中的多环现象. 国内的柯映林等<sup>[4]</sup>提出了基于点集的“紧包围盒”分割来建立散乱点之间的邻接信息, 构建平面切片数据, 并根据散乱点间的拓扑关系, 重建平面曲线. 吴杭彬等<sup>[5]</sup>根据点云数据的特点, 提出了点云数据的分层模型, 实现了等值线的分层提取, 并根据等值线的细节程度不同, 对等值线进行分类, 实现了多细节表达.

以上方法, 通过集成 RE 和 RP, 避免了曲面拟合和 STL 文件生成的过程<sup>[6-7]</sup>, 其应用领域多为机械制造, 目的是把曲线模型作为 RP 系统的输入. 本文结合逆向工程和快速成型两种技术, 针对海量散乱点云, 提出一种基于切片的海量点云数据轮廓特征线快速生成算法, 并通过实例分析了算法的效率

收稿日期: 2011-08-05

基金项目: 国家自然科学基金(40971241); 国土环境与灾害监测国家测绘局重点实验室项目(LED2011B01)

第一作者: 程效军(1964—), 男, 教授, 工学博士, 博士生导师, 主要研究方向为数字近景摄影测量与精密工程测量.

E-mail: cxj@tongji.edu.cn

通讯作者: 贾东峰(1985—), 博士生, 主要研究方向为三维点云数据的处理和应用. E-mail: sosojd7@163.com

和可行性.

## 1 算法描述

### 1.1 点云切片的生成

设有散乱点集  $S = \{p_1, p_2, \dots, p_n\}$ ,  $p_i = \{x_i, y_i, z_i\} \in \mathbf{R}^3$ , 则点云切片的生成可描述为: 采用一组平行平面沿给定方向对三维点云进行划分, 并将三维点集转化为二维切片数据. 点集  $S$  的坐标范围为  $(x_{\min}, y_{\min}, z_{\min}) \sim (x_{\max}, y_{\max}, z_{\max})$ , 有一平面集  $T$ , 平行于  $xoy$  平面, 法矢  $n$  指向  $Z$  轴正方向, 该平面集  $T$  由一组  $Z$  坐标序列表示:  $Z = (Z_0, Z_1, \dots, Z_n)$  且  $Z_0 < Z_1 < \dots < Z_n$ , 其中:

$$\begin{cases} Z_0 = Z_{\min} \\ Z_n = Z_{\max} \\ Z_k = Z_0 + iZ_{\text{pitch}}, \quad i = 1, 2, \dots, n-1 \end{cases} \quad (1)$$

式中,  $Z_{\text{pitch}}$  是分层厚度. 本文通过获取对象的高度  $H$ , 计算分层厚度. 为了控制切片中投影带的宽度, 点云的分层厚度计算如下:

$$Z_{\text{pitch}} = H \frac{1}{10k}, \quad k \in [4, 8] \quad (2)$$

通过定义参考平面位于分层数据的中间位置, 将每层数据向参考平面投影即可得到切片数据.

### 1.2 特征点提取

本文采用基于数字图像的方法提取切片数据的特征点, 该方法由 Chen 等<sup>[8]</sup>和 Zhang 等<sup>[9]</sup>提出. 首先建立一个栅格化的数字平面(取参考平面), 将每层点云数据投影到数字平面上, 对于落入点的栅格赋予值“1”(黑块), 对于未落入点的栅格赋予值“0”(白块). 如图 1a 为将点云数据投影到栅格化的数字平面; 图 1b 为投影到数字平面的第 24 层切片.

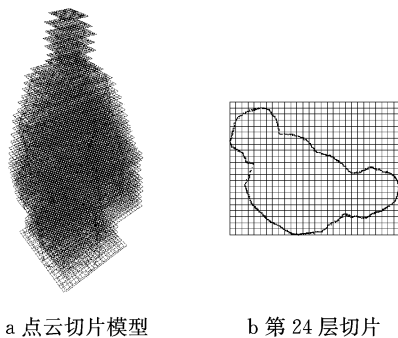


图 1 将空间点投影到栅格化的数字平面

Fig.1 Projecting spatial points onto digital grid plane

文献<sup>[3, 6, 10]</sup>采用基于细化模板的特征点提取算法来提取投影后的数据点. 该方法由于未考虑栅格的边长对特征提取的影响, 会造成特征点的删除,

因此使用上述方法提取特征点后, 需采用一个基于最大距离误差的算法来判断删除的冗余点中是否含有特征点, 并根据距离判断以恢复删除的特征点, 该过程会影响特征点提取的效率.

为了提高平面轮廓特征点提取的效率, 在建立栅格化的参考平面时, 可以通过讨论栅格边长对特征提取的影响, 提高轮廓特征点提取的效率, 该方法的具体过程如下.

#### 1.2.1 数字平面栅格边长的计算

平面栅格边长的大小直接影响到特征点的提取和特征线的构造精度. 为了减少平面栅格边长对特征提取的影响, 提高算法的稳健性, 本文采用以下方法计算平面栅格的边长.

首先, 计算平面点集的最小包围盒, 获取平面点集的坐标范围  $(x_{\min}, y_{\min}) \sim (x_{\max}, y_{\max})$ , 栅格的边长  $s_{\text{size}}$  计算公式如下:

$$s_{\text{size}} = \alpha \sqrt{\frac{(x_{\max} - x_{\min})(y_{\max} - y_{\min})}{n}} \quad (3)$$

式中:  $\alpha$  是尺度因子, 用于调节栅格边长的大小,  $n$  是点的个数. 文献<sup>[11]</sup>讨论了  $\alpha$  的最佳取值范围是  $[1, 1.5]$ , 本文取  $\alpha$  的值为 1.2.

#### 1.2.2 特征点的提取

将空间点投影到栅格化的参考平面, 以栅格中的点云数据重心取代栅格内的点作为特征点, 则第  $i$  个栅格内的特征点的坐标计算公式为:

$$\bar{x}_i = \sum \frac{x_i}{n} \quad (4a)$$

$$\bar{y}_i = \sum \frac{y_i}{n} \quad (4b)$$

如图 2a 为将空间点投影到按照式(3)计算的栅格平面中; 图 2b 为提取每一个栅格中的点云重心作为轮廓特征点.

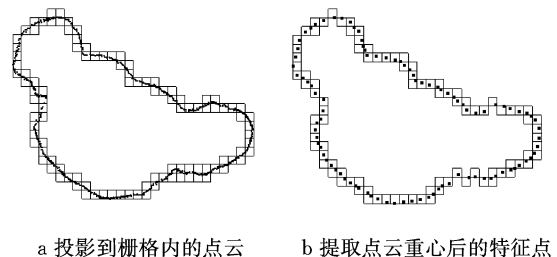


图 2 特征点的提取

Fig.2 The extraction of feature points

### 1.3 轮廓特征线的生成

如何根据提出的特征点快速构建轮廓特征线是本文研究的重点, 由于分层厚度和栅格边长的设置可以有效地控制切片投影带的宽度, 根据平面栅格

中投影带的特点,本文提出了双向索引连通法来快速构建特征线,方法原理介绍如下.

### 1.3.1 双向索引连通法

首先按 X 方向建立栅格的索引,判断栅格内是否含有特征点,分层构造特征线段;再按 Y 方向建立栅格索引,判断栅格内是否含有特征点,分层构造特征线段,图 3 所示是按双向索引连通法构造特征线的原理图;图 3a 显示的是栅格平面的特征点;图 3b 和图 3c 是分别按 X 和 Y 方向根据栅格的索引分层连接特征点;图 3d 是轮廓特征线的生成图.

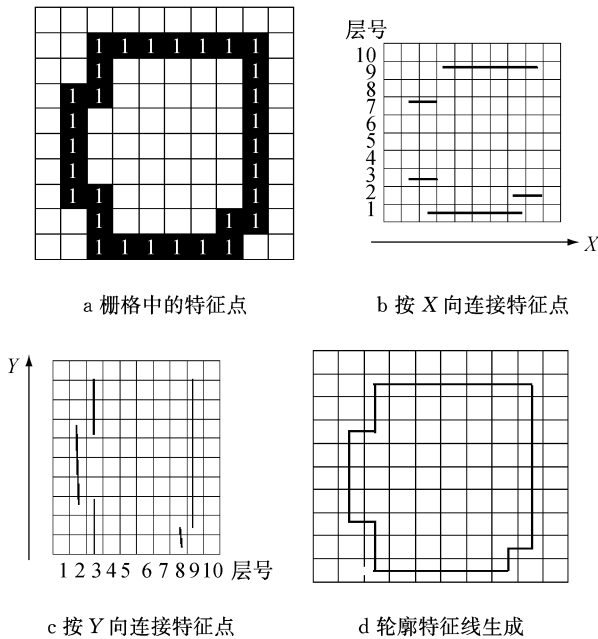


图 3 双向索引连通法生成轮廓特征线

Fig.3 Contour generation using bidirectional index connecting method(BICM)

### 1.3.2 对角邻域、田字形邻域以及邻域缺失的连通

基于双向索引连通法能快速连接特征点的四邻域,但是对于如图 4 所示的对角邻域、田字形邻域及邻域缺失的情况造成轮廓特征线的断裂和偏转,需要设计特定的判断条件进行连通. 图中,  $P, p_i$  表示点位.

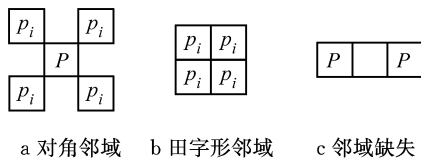


图 4 几种邻域类型

Fig.4 Several types of neighborhood

首先根据 X 方向的索引存储每层特征线段的的首尾特征点(端点),判断这些端点含有四邻域点的个数. 文中采用的存储容器为 C++ 标准模板库中的

vector 容器.

**对角邻域的连通:**从端点容器中任意取出一点,如果其含有两个四邻域点,则从容器中删除该点;如果小于等于一个四邻域点则保存该点到对角邻域容器,循环判断直到遍历所有端点. 判断完毕,从对角邻域容器中任意选出一,根据索引判断特征点容器中是否含有该点的对角邻域点,如果有则连接两点,然后从容器中剔除这两个点;如果没有对角邻域点,则保存该点到邻域缺失容器.

**邻域缺失的连通:**经过对角邻域的连通,容器中剩余的点为邻域缺失的点,任意取出一点,判断该点到其他端点(可选择相邻层的端点)的距离,由于切片数据存在不闭合情况,如建筑点云切片的门窗处形成点云的断裂,要设定距离阈值进行判断,本算法设置  $2s_{size}$  作为距离阈值,如果端点距离大于距离阈值则认为自然断裂无需连接,如果小于距离阈值则选择并连接离该点最近的端点,完成邻域缺失的连通.

**田字形邻域的连通:**为了准确找到田字形邻域的位置,首先需要判断特征点的邻域关系,如果点  $P$  存在如图 5 所示的情况( $Y_{num}$  为 Y 方向上的序号),则存储该点及其邻域值,循环判断直到遍历所有特征点;找到田字形邻域后,需要对田字形邻域进行合并,本文采用均值合并的方法对其进行合并,由于可按上下邻域取平均或左右邻域取平均,算法首先分别按两个方向取平均,判断某方向合并后的均值点是否存在两个四邻域点,如果存在则用其替代“田”字邻域中的 4 个点,并删除容器中的四邻域点,然后对均值点进行连接.

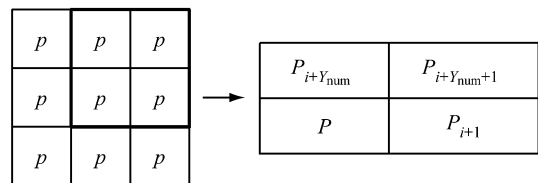


图 5 田字形邻域的判断条件

Fig.5 The judging condition of “田” shaped neighborhood

## 2 实例分析

本文在 Visual C++ 6.0 环境下结合 COIN3D 库测试上述算法,测试用的点云数据,如图 6 所示,图 6a 为标准构件的点云数据(点云个数 4102);图 6b 为建筑物的点云数据(点云个数 1 985 617);图 6c

为华佗雕像的点云数据(点云个数 708 987). 为了说明算法的可行性, 本文针对以上 3 种点云数据进行实验.

第一组实验: 测试算法对对角邻域、田字形邻域以及邻域数据缺失的连通情况. 如图 7a 所示为标准构件的第 2 层点云切片; 图 7b 是点云切片生成轮廓线过程中的对角邻域造成了轮廓线的断裂; 图 7c 是根据本算法提出的方法对对角邻域进行连通, 从图中可以看出算法判断出了对角邻域点并对其进行了连通.

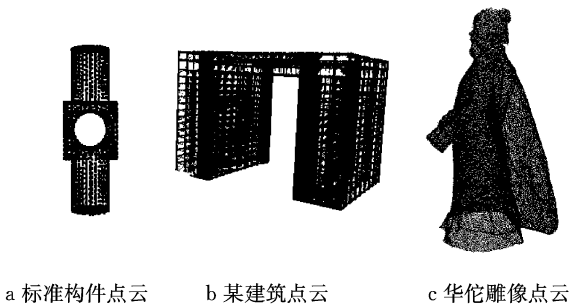


图 6 点云数据  
Fig. 6 Point cloud

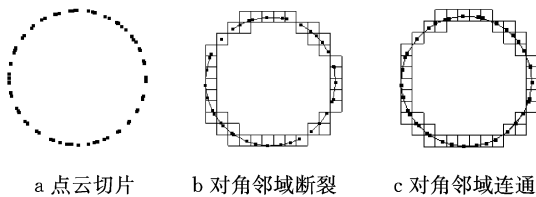


图 7 对角邻域的连通

Fig. 7 The connecting of diagonal neighborhood

图 8a 所示为标准构件的第 5 层点云切片; 图 8b 是轮廓线生成过程中的邻域缺失造成轮廓线的断裂; 图 8c 是根据本算法提出的方法实现了邻域缺失情况的连通.

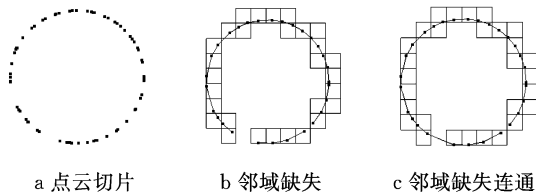


图 8 邻域数据缺失的连通图

Fig. 8 The connecting in case of missing neighbor data

图 9a 所示是华佗雕像的第 16 层点云切片; 图 9b 是根据本算法计算并提取切片数据的特征点; 图 9c 由于田字形邻域造成轮廓线的局部偏转; 图 9d 根据本算法判断出田字形邻域的位置; 图 9e 根据均值合并的方法对田字形邻域进行合并; 图 9f 轮廓线的生成, 可以看出本算法已经消除了田字形邻域造成

的轮廓线偏转, 实现了轮廓线的连通.

第二组实验: 本算法生成的轮廓线和逆向工程软件 Geomagic 以及导入计算机辅助设计软件 (CAD) 手绘产生的轮廓线效果进行比较. 如图 10a 所示是某建筑物的第 7 层点云切片; 图 10b 是根据本文提出的算法生成的轮廓线, 准确表达了点云切片数据的轮廓特征; 图 10c 是逆向工程软件 Geomagic 生成的轮廓线, 图中有许多断裂; 图 10d 是将点云切片导入 CAD 后通过手工绘制的轮廓线, 虽然可以表达点云切片的轮廓特征, 但是人工判断以及手工连接需要花费大量的时间. 通过比较可以

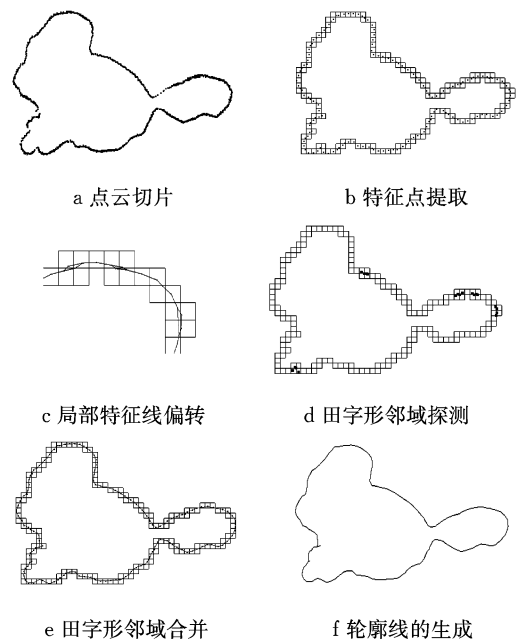


图 9 田字形邻域的连通

Fig. 9 The connecting of "田" shaped neighborhood

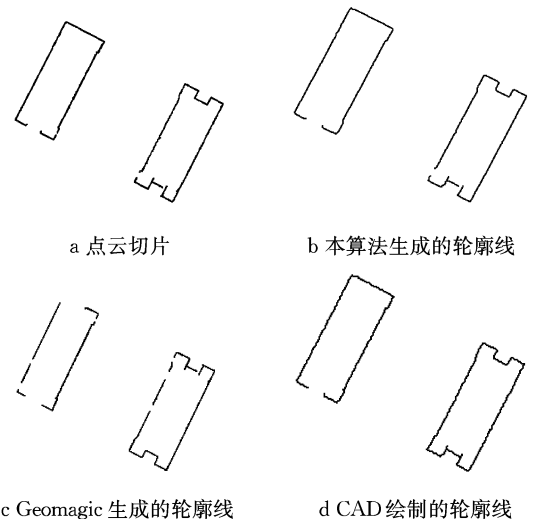


图 10 轮廓线的生成效果比较

Fig. 10 The comparison of contour generation of different methods

看出本算法比其他商业软件可以更准确地表达出点云数据的轮廓特征。

第三组实验,本算法和凸包算法的时间复杂度分析比较. Graham<sup>[12]</sup>提出的平面点集凸包 Graham 扫描算法,其时间复杂度为  $O(n \lg n)$ . 文献[13]提出了平面点集凸包的最优实时算法,使其复杂度达到了  $O(n)$ . 本算法根据 X、Y 双向搜索连接,并对局部的连通性进行判断,其时间复杂度是线性的,可以达到  $O(n)$ . 图 11 是采用本算法生成的雕像点云的轮廓线模型. 图 11a, b, c 分别是华佗点云的正视、前视、俯视轮廓线模型。

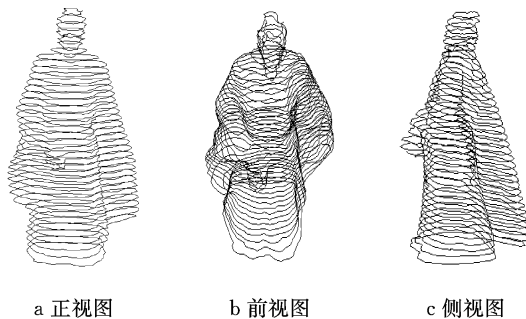


图 11 华佗轮廓线模型

Fig. 11 The contour model of Huatuo statue

### 3 结论

在针对海量散乱点云的算法中,基于点云特征的表面重建是近年研究的热点,如何从海量散乱点云数据中提取特征点构造特征线,并基于特征线实现海量点云的表面重建是算法研究的关键。

本文通过结合逆向工程和快速成型技术提出了一种快速构造点云轮廓特征线的方法. 该方法:(1)采用数字图像的方法,通过建立数字栅格平面提取点云切片的特征点;(2)量化了数字平面的栅格边长并以落入栅格中的点云重心作为特征点;(3)基于特征点在栅格中的索引值,提出了双向连通索引法快速构造点云特征线. 通过和商业软件以及凸包算法的分析和比较,本算法可以快速、准确地生成点云的轮廓曲线模型. 进一步研究的重点是讨论栅格边长的大小和曲线模型的精度的关系,并建立二者之间的数学模型。

### 参考文献:

- [1] Lee K H, Woo H. Direct integration of reverse engineering and rapid prototyping [J]. Computer Industry Engineering, 2000, 38:21.
- [2] WU Y F, WONG Y S, Loh H T, et al. Modeling cloud data using an adaptive slicing approach [J]. Computer-Aided Design, 2004, 36:231.
- [3] Liu G H, Wong Y S, Zhang Y F, et al. Error based segmentation of cloud data for direct rapid prototyping [J]. Computer-Aided Design, 2002, 35:633.
- [4] 柯映林,王青. 反求工程中的点云切片算法研究[J]. 计算机辅助设计与图形学学报, 2005, 17(8):1798.  
KE Yinglin, WANG Qing. Research on point cloud slicing technique in reverse engineering [J]. Journal of Computer-Aided Design & Computer Graphics, 2005, 17(8):1798.
- [5] 吴杭彬,刘春. 激光扫描数据的等值线分层提取和多细节表达[J]. 同济大学学报:自然科学版, 2009, 37(2):267.  
WU Hangbin, LIU Chun. Point cloud-based stratified contour extraction and its multi-LOD expression with ground laser range scanning [J]. Journal of Tongji University: Natural Science, 2009, 37(2):267.
- [6] Kumbhar V K, Pandey P M, Rao P V M. Improved intermediate point curve model for integrating reverse engineering and rapid prototyping [J]. The International Journal of Advanced Manufacturing Technology, 2007, 37: 553.
- [7] YIN Zhongwei. Direct integration of reverse engineering and rapid prototyping based on properties of NURBS or B-spline [J]. Precision Engineering, 2004, 28:293.
- [8] CHEN Y H, NG C T, WANG Y Z. Data reduction in integrated reverse engineering and rapid prototyping [J]. International Journal of Computer Integrated Manufacturing, 1999, 12:97.
- [9] ZHANG Y F, WONG Y S, Loh H T, et al. An adaptive slicing approach to modeling cloud data for rapid prototyping [J]. Mater Process Technology, 2003, 140:105.
- [10] Jang B K, Chin R T. Reconstructable parallel thinning [J]. International Journal of Pattern Recognition Artificial Intelligence, 1993, 7(5):1145.
- [11] Piegel L A, Tiller W. Algorithm for finding all  $k$  nearest neighbors [J]. Computer-Aided Design, 2002, 34:167.
- [12] Graham R L. An efficient algorithm for determine the convex hull of a finite linear set [J]. Information Processing Letters, 1972(1):132.
- [13] 王志强,洪嘉振,肖立瑾. 平面点集凸包的最优实时算法[J]. 计算机学报, 1998, 8(21):351.  
WANG Zhiqiang, HONG Jiachen, XIAO Lijin. An optimal real time algorithm for determining the convex hull of a set of points in a plane [J]. Chinese Journal of Computers, 1998, 8(21): 351.