

首都师范大学

论文分类号：TP391.72

硕士学位论文

密 级：无

光栅图像矢量化技术研究

Study on the Vectorization of Raster Images

专 业：计算机应用技术

研 究 方 向：多媒体技术应用

研 究 生 姓 名：刘 玉 兰

指导教师姓名：葛 庆 平

指导教师职称：副 教 授

2005 年 4 月

摘 要

光栅图像的矢量化主要由图像分割及矢量获取两部分组成。通过图像分割得到目标图像的边缘线或骨架线,并以离散点序列形式表示出来。矢量获取则将这些点序列转换成首尾相连的曲线段或具有一定逻辑关系的图元对象,如圆弧、直线、椭圆弧等。矢量获取是光栅图像矢量化的关键部分,也是本文的研究重点,本文在后面提到的矢量化算法,都是指矢量获取算法。衡量一个矢量化算法效果好坏的根本标准是曲线段或图元对象与原始点序列间的误差大小及其相对原始数据的压缩率。从理论上讲,误差越小,压缩率越大,则矢量化的效果越好。但这种高压缩率和低误差往往是以大量的计算为代价换来的,如在某些最优化算法中,通过不断排除不满足要求的解,最后得到具有最优性能的解,此过程需要进行大量的计算,不符合实际生产中高效率的要求。在实际的生产应用中,往往是一定的精度和压缩率就能满足生产需要,因此应当根据具体的要求选择合适的矢量化算法,避免不必要的计算。

本文研究的课题背景是鞋样设计和扩缩 CAD 系统(在后面简称为鞋样系统)。该系统将放有鞋样的背景图片由数码相机输入到计算机内,经图像分割后得到鞋样轮廓的数字点序列表示。由于这些离散的点序列数据量太大且存在大量细小的毛刺,不利于设计人员对其进行编辑和利用切割机进行切割,因此需转换成矢量图形。根据实际需要,本系统中拟合曲线与鞋样边缘的最大误差不能超过 2 毫米(对应屏幕中的 5 个像素点),且矢量图在保持原始边缘特征的基础上能对原始数据能进行较大的压缩。因此,本文在研究现有算法的基础上提出了一种分段递归拟合矢量化算法。该算法首先通过计算曲率提取尖点将轮廓进行分段,然后对每两个尖点之间的曲线段用抛物样条曲线进行递归拟合。在提取尖点时又提出了一种间接的方法,大大简化了曲率的计算。最后实验证明,递归方法能通过中止条件很好的控制拟合的精度和压缩率,在满足精度要求的同时减少不必要的计算。因此该算法能满足鞋样系统的要求。

关键词: 矢量化, 矢量获取, 图像分割, 曲线拟合, 光栅图像

Abstract

The vectorization of raster images including two steps: image division and vector getting. Through image division we get the edge or framework of a target in form of point series. Vector getting fits the point series with curves such as arcs, lines and ellipse arcs etc. These curves are joining end to end. Vector getting is a key step in the process of raster image vectorization and is the most important part of this article. In the following part of this article vectorization is equal to vector getting. A good vectorization algorithm can not only fit the point series with little error, but also can reduce the data in large amount. In theory, the less fitting error and bigger compress rate, the better a vectorization algorithm is. But little fitting error and big compress rate always take a large amount of calculating, such as in some optimum algorithms, to find the best result always need many times of iterative calculating. In practice, certain fitting error and compress rate can satisfy the demand, so we should choose a suitable algorithm to avoid unnecessary calculating.

The background of this article is the shoe pattern design and enlarge and constrict system (shoe pattern system in abstract). In this system, photos of shoe patterns are transmitted to computer from digital cameras. After image division, we get the outlines of shoe patterns in form of point series. The amount of points of a shoe pattern is very large and there are many noises among these points, these make us difficult to enlarge or constrict the shoe pattern, so these point series should be turned into vectors before a shoe pattern is operated in computer. Fitting error of vectorization should be less than 2mm(5 pixels in the digital image) in this system, and a certain compress rate should be get in condition of keep the feature of the shoe pattern outline. In this article, we present a recursive vectorization algorithm based on the existing ones. This algorithm first separate the outline with cusps and a new way of getting cusp points is presented. This new cusp getting way has a less amount of calculating compared with general ways. After separating, algebraic curves was used to fit digital curves between every two cusps in recursive. The experiment of this algorithm shows that recursive algorithm can control the fitting error and compress

rate very well through ending condition, without unnecessary calculating, so this algorithm can satisfy the demand of the shoe pattern system.

Keywords: vectorization, vector getting, image division, curve fitting, raster image

声明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已发表或撰写过的研究成果，也不包括为获得首都师范大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

签名： 日期： 年 月

目 录

第 1 章 概述	1
1.1 矢量化研究背景及意义	1
1.1.1 背景	1
1.1.2 基本概念及研究意义	1
1.2 矢量化研究现状分析	3
1.2.1 总体形式分析	3
1.2.2 现有算法分析	4
1.3 本文的主要研究内容	7
第 2 章 鞋样设计和扩缩 CAD 系统	8
2.1 鞋样设计和扩缩 CAD 系统简介	8
2.2 矢量化与鞋样系统	9
第 3 章 图像分割	11
3.1 门限处理	11
3.2 噪声去除	13
3.3 边缘跟踪	13
3.3.1 像素的邻接性	14
3.3.2 像素的连通性	14
第 4 章 边缘曲线的分段拟合	17
4.1 基本概念	17
4.1.1 插值、逼近和拟合	17
4.1.2 误差计算	18
4.1.3 算法的评价标准	19
4.2 尖点提取	20
4.2.1 算法思想	22
4.2.2 算法数学基础	23
4.2.3 算法设计	24
4.2.4 实验结果及分析	25
4.3 曲线拟合	27
4.3.1 抛物样条曲线	27
4.3.2 现有拟合算法	28
4.3.3 递归算法	29
4.3.4 算法分析及试验结果	30
4.4 本章小结	37
第 5 章 结论	38
参考文献	39
攻读硕士学位期间参与的学术活动	42
一、发表的学术论文	42
二、参加的学术研讨会	42
三、参与的主要科研项目	42
致 谢	43

第1章 概述

1.1 矢量化的研究背景及意义

1.1.1 背景

自从工业文明诞生以来,图纸作为工业设计和工业规划的主要媒介在工程领域充当着不可替代的角色。而随着计算机技术的发展,工业设计和规划正在告别手工绘图时代,进入计算机辅助设计和制造时代(CAD\CAM)。计算机辅助设计与制造技术是近年来工程技术领域中发展最迅速、最引人注目的一项高级技术,它已经逐步成为工业生产现代化的重要标志之一。该技术的准确性、智能性和易编辑性大大的缩短了产品开发、模具设计和加工的周期,提高了产品质量、降低了成本、增强了企业市场竞争能力,得到工程界的广泛认可。现在计算机辅助技术已经渗透到工程设计及工业生产的各个阶段,从图纸设计到信息提取,从自动读图、理解到自动生产,计算机技术与工程行业正在紧密结合,向支持工业全过程自动化的目标迈进。但是由于大多数工程自动化软件只能存储、编辑和管理CAD图形文件,而计算机直接得到的图片通常是以光栅图像的形式存在,这就需要运用计算机技术将光栅图像自动转换为CAD图形文件,矢量化技术就是在这种形式下发展起来的。国内外在矢量化方面的研究已经进行了许多年,但由于输入图像本身的复杂性以及在输入过程中引入的噪音的干扰,目前自动转换的正确程度和速度尚不能令人满意,因而仍有较大的研究空间。

1.1.2 基本概念及研究意义

计算机中图像文件的格式主要有两大类:一类是位图文件格式,另一类是矢量图文件格式。位图也称光栅图或点阵图,它把一幅图分成许多栅格图素,称每一个栅格为一个图素点,简称像素(pixel)。位图就是一组描述每个图素点的数据。位图图像文件格式适用于有复杂颜色和灰度级变化的图,如照片、绘画、数字化视频图像等。

矢量图用几何图形的特征数据及其属性来描述图像,如一条线用其两个端点的坐标来描述,一段圆弧用其起点和终点坐标、圆心坐标及半径来描述。矢量图的最小单位是图元,如直线、圆弧、椭圆弧等基本图形。矢量图图像文件格式适用于线性图,如机械图形、建筑图形、地理等高线图。CAD/CAM系统中所用的

都是矢量图。线性图用矢量图描述的数据量比用光栅图描述的小得多。

对光栅图像进行分析、识别并重建其中图形对象的过程称为矢量化^[1]。光栅图像以像素为表示单元，因此对图像的编辑只能通过像素级操作进行，不易进行各种变换，如图像的旋转、缩放等。并且以像素表示的图像中往往存在大量的冗余信息，存储量过大，例如一张 A0 尺寸的工程图纸用 300dpi 的分辨率、单色扫描后得到的图像文件大小在 15MB 左右，这对一般的图纸管理系统是一笔很大的开销。而矢量图用直线、圆弧等基本几何图形来描述，因此只需对这些几何图形的参数进行存储，例如直线只需存储其端点坐标，圆弧只需存储其圆心坐标、半径、起点和终点，这样就可以在不损失图像信息的前提下大幅度减少存储空间，还能灵活高效的对图像进行编辑，因此研究光栅图像的矢量化具有很高的实际应用价值。

光栅图像的矢量化目前主要应用在工程图纸的打描识别技术中，其实际的应用主要表现在以下几个方面：

- **建立工程图档案数据库：**在使用计算机进行辅助设计之前，工程图纸都是人工绘制的，在计算机应用日益普及的今天，急需将这些图纸都送入到计算机中进行矢量化和识别，以得到进一步的管理和使用。
- **对已有的图纸进行分析或编辑：**在计算机辅助设计技术中，将现有的图纸送入计算机，并在此基础上加以局部修改产生新的设计，可以大大提高设计效率，降低设计成本。又如 GIS 技术中，由于矢量数据可以更精确定义位置、长度和大小，因此矢量化对于 GIS 数据的分析和变换、空间数据拓扑结构分析以及空间数据与属性数据的联合分析都是必不可少的^[2]。
- **作为形体建模的一种工具：**随着计算机技术不断向智能化方向发展，人们提出了让计算机在图形识别技术的基础上，结合领域知识自动理解工程图的要求。进一步，在自动读图的基础上，建立图与图之间的关联，从而实现由多张二维视图重建三维模型的目标。

从以上分析可以看出，通过将直观而缺乏逻辑性的大量点阵数据转化成少量的比较抽象而富于逻辑性的矢量图，可供当前多数的 CAD/CAM 系统使用，并促进产品设计和制造的自动化。

1.2 矢量化研究现状分析

1.2.1 总体形式分析

国内外矢量化的研究从上个世纪 70 年代开始,其最初的应用范围很窄,主要用来分析地图;方法也很单一,通常采用细化算法或轮廓检测算法加以简单的线条拟合,得到矢量结果。

80 年代中后期, CAD 技术在工业设计领域得到了广泛的应用。但是,由于长期以来工业设计都使用手工绘图,使得以往大量的工程图都以纸制图片的形式保存、管理和使用着。这些纸制工程图往往是一个系统或行业长时间积累下来的,它们完整的保存了以往的宝贵经验,对以后的设计有很好的借鉴作用。因此为了能在 CAD 系统中修改和重复使用这些图纸,需要将它们转换为 CAD 系统能处理的格式,即矢量图格式。将纸制工程图纸转换为用计算机管理的矢量图并建立相应的数据库,可以减少图纸更新和维护所耗费的人力资源,并使其更加便于存储、修改、查询和交流。但是实际当中用扫描仪或数字化仪直接将工程图纸输入到计算机后得到的都是光栅图像,光栅图像的数据量非常大,且无法在 CAD 系统上直接进行编辑,因此需要对其进行矢量化。通过对光栅图像进行矢量化,能极大的提高产品的设计效率,缩短生产周期,促进工程设计、产品设计和制造大现代化,从而引发了矢量化研究的高潮。

对扫描工程图纸的矢量化引发了矢量化的研究高潮,并逐步渗透到社会生产的各个领域。到了 90 年代矢量化已广泛应用于如地理信息系统、工程业、房地产业及工业制造等行业,同时也出现了很多商品化的通用矢量化软件,典型的如德国 Softlec 公司的 VPStudio、挪威 Rasterex 公司的 RxSpotlight、美国 GTX 公司的 GTXRasterCAD PLUS 等等,这些软件都能对光栅图进行矢量编辑或进行自动矢量化。另外,由于不同领域的特定要求,还出现了一些专业化的矢量化软件,如主要针对地图和 GIS 领域的 Able 公司的 R2V,针对鞋样设计的法国三维立体浮雕软件 TYPE3。这些系统都具有完善的处理功能和易用的用户接口,但是在矢量化精度和速度上尚不能完全达到工程自动化的需要,普遍具有对噪音、缺损敏感等缺点。

随着计算机技术的发展,设计和生产的自动化是一种必然的发展趋势,因此开发出能够实现由光栅图像到矢量图形的快速转换的各类行之有效的应用程序,

对社会生产的发展具有重要的意义。但是工程图像的智能识别与自动输入因涉及到计算图形学、图像处理、模式识别、人工智能等多种技术,虽然在学术界和工程界受到普遍关注,目前仍然是CAD领域中的一个“瓶颈”问题。其需要解决的一个关键问题就是图像的矢量化。

1.2.2 现有算法分析

矢量化用基本图元(线段、圆弧和样条曲线等)来替代光栅图中的点序列,实现方法有很多,但总的来说可以分为两大类:一是直接从轮廓中抽取基元,即恢复优先的方法;二是先抽取基元之间的分界点,然后再逐一用矢量基元来进行拟合,即分割优先的方法。

1. 恢复优先的方法:

恢复优先的方法又可以分为基于细化的算法、基于游程编码的方法、基于轮廓的方法等。

基于细化的算法^[2-9]:基于细化的算法将图像中所有像素点分为两类:轮廓像素和非轮廓像素。通过不断反复删除轮廓像素,最后得到1个像素宽的图像骨架,然后通过折线段逼近将骨架点序列转换成低级矢量格式(直线段或折线段),最后用基于矢量的图形识别方法在短线段中识别出其中的图形对象。

基于游程编码的算法^[10-13]:游程编码定义为点阵图像在线段识别以前的一种半矢量表示法。它的两个端点为图像轮廓点,长度表达图形线宽,而不同游程编码间的分裂、合并关系即体现像素连续性,因此,游程编码能够代表图像轮廓,并直接综合轮廓两端信息。基于游程编码的算法通常先对图像进行编码,然后分析行段生成各种形式的邻接图结构,由图像中线状区域的行段中点拟合而成的线段作为邻接图中的边,非线状区域(相交或粘连处)作为邻接图中的节点连接相邻的边;最后在图结构上进行线段延伸、合并识别图形对象。

基于像素跟踪的方法^[14]:通过像素级跟踪获得图像的中轴线。如文献[4]中提出的稀疏像素跟踪法,它可从任意可靠中点开始按预定步长沿水平或垂直方向进行跟踪,获得每一步的中点,跟踪完整个图像后,对所得的中点序列加以折线化去除冗余点得到图像的中轴线表示,最后在轴线的基础上识别图形对象。

综合以上几种方法可以看到,恢复优先的方法其过程可以用下图来表示:

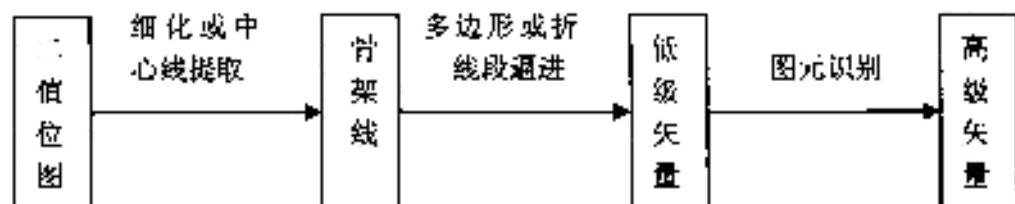


图 1.1 恢复优先矢量化方法的流程

由上图可以看出，该方法主要分三个步骤进行：首先通过细化或中心点提取算法获得整个图像的骨架点，然后通过多边形拟合或折线化等处理生成图像的低级矢量形式，最后采用基于矢量的图形识别方法在短线段或折线段中搜索、拼接出图像中原有的图形对象，从而生成图像的高级矢量形式。该方法有效的将图形对象识别与像素级操作分离，减少像素存取次数，但还存在以下缺陷：

(1) 细化算法容易遭受噪声侵蚀，并且在结点处会产生畸变。若骨架像素信息充足，则可归纳、总结畸变规律，进行校正，如文献[8]、[9]；但若图像稍微复杂，则难以进行校正。此外，由于细化算法在像素层进行局部操作，并反复进行，因此计算量与内存需求都非常巨大，难以达到实时要求。

(2) 几乎所有的细化算法和中轴线提取算法都不能正确处理图形对象的相交情况，常将对应于一线段的位图识别成多条短线段。这就导致了第三步中为合并所需的大量短线段搜索、测试和拼接操作，并且多段之间由于噪音、缺损等造成的方向不一致性给拼接操作带来困难。矢量化的速度和精度都因此受到影响。

(3) 现有图形识别方法很少讨论如何处理已识别图形对象对应的低级矢量数据，合理的推测是标记这些数据而非删除，因为某些相交或粘连部分可能对后继识别仍然有用。然而，这也意味着相交或粘连的图形对象之间的相互干扰在整个矢量化过程中一直存在。因此一些重要的处理在这种数据环境下工作就会很复杂并且容易出错。

2. 分割优先的方法：

分割优先的方法首先取基元之间的分界点，然后根据分界点确定拟合曲线的参数进行拟合。分界点是人类认知形状的重要内容，可以用来描述、识别和匹配形状。经过多年的研究，已经提出多种分界点检测算法：

1. 曲率计算^[15-17]，根据曲率（夹角）大小来提取分界点，计算量相对较小。但抗噪性差。

2. 小波变换^[18-20]，基于小波变换检测角点，抗干扰能力强，但计算较为复杂。

3. 模糊推理，根据人对局部图形特征的认识，用模糊推理来检测角点。

4. 优化方法^[21-23]，基于遗传算法来选择分界点进行曲线拟合。即根据拟合基元的特征确定一组最佳分界点，使得拟合误差和点压缩率都满足一定的条件。这种方法的缺点是不适用于长的曲线且计算复杂。

一般情况下，单独使用以上的某一种方法很难达到很好的矢量化效果，因此通常都是把两种或两种以上的方法结合起来使用。如文献[29]提出了一种基于预选关键点的遗传算法，即先利用曲率提取出一部分点作为初始分界点，然后再利用遗传算法在这些初始点中进行最终分界点的选取。

分割优先的矢量化方法其过程可以用下图来表示：

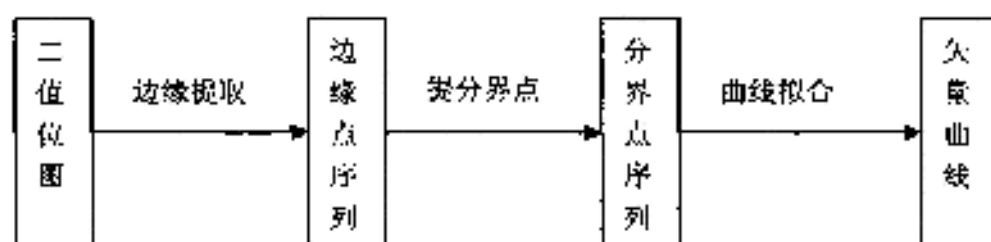


图 1.2 分割优先矢量化方法的流程

该方法也主要分三个步骤进行：首先对二值化后的光栅图像进行边缘跟踪得到图像轮廓的点序列表示，然后提取出图像轮廓的分界点。最后利用分界点计算出拟合曲线的参数的到矢量曲线。分割优先矢量化方法比较适合于图像形状变化多样、难以在其中识别出具体的基本图元的情况，如本论文在后面将要讨论的鞋样形状。但是目前该方法还存在的问题是：

(1) 用作拟合基元的代数曲线形式比较复杂，给算法的设计带来难度，并且计算量很大。如文献[24]中用直线和圆弧进行迭代拟合，每次迭代计算误差时需要分别对圆弧和直线的拟合误差各进行一次计算，大大增加了计算量。又如用三次 B 样条曲线计算一个点的坐标需要进行一次立方运算、一次平方运算及二次矩阵运算，在计算拟合误差时计算量非常大。

(2) 分界点的提取没有考虑到拟合的误差或计算量太大。在上面几种分界点提取算法中，前 3 种方法只考虑到轮廓某一方面的特征，因此通过这些分界点对整体曲线进行拟合时很难达到要求的精度。只有优化算法把误差作为分界点选

取的条件，因此能达到较高的精度。但该方法往往需要进行多次迭代，且每次迭代需要对整体误差进行一次计算，因此计算量太大，算法效率不高。

1.3 本文的主要研究内容

本论文的主要研究任务是光栅图像的矢量化技术。矢量化的目的就是要将光栅图文件转换成 CAD 系统能够接受的矢量图文件，使其在 CAD 系统里可以直接编辑、修改和输出。一般来讲，一个矢量化系统由以下几个基本部分组成：

1. 图像的获取
2. 图像分割
3. 矢量转换
4. 矢量编辑及输出

从矢量转换的角度来分析，可以将整个系统划分为三个部分：前处理、矢量化及后处理。前处理以图像处理为主，如图像分割。后处理则以图形处理为主，如矢量编辑。在从图像到图形的转换过程中，矢量化转换是关键的一步。矢量转换结果的好坏，往往跟采用的矢量化算法及图片自身特点有很大关系，本文的重点是对矢量转换算法的分析。前面介绍了两类矢量化算法，由于分割优先的方法比较适用于本文所要处理的问题，因此本文讨论的是分割优先的算法。

整个文章内容安排如下：第一章是关于矢量化的概述，重点分析了现有矢量化算法及其特点。第二章介绍本文研究的项目背景。第三章介绍矢量化前处理：图像的分割。第四章详细介绍本文采用的矢量化算法，是本文的重点，也是本文的创新所在。第五章为结论部分，是对本文内容的一个总结。

第2章 鞋样设计和扩缩 CAD 系统

2.1 鞋样设计和扩缩 CAD 系统简介

随着现代工业的发展和水平的提高,使得人们对皮革制品的需求不断增加。然而,由于皮革自身特性的限制,传统的皮革生产加工工艺的效率低下,并且加工成本高昂,成为了皮革制品生产发展的一个重要的障碍。随着计算机技术的飞速发展,将计算机辅助技术与传统的皮鞋生产工艺相结合成为了一个必然趋势。

在 CAD 技术用于皮鞋制造业之前,皮革鞋样的切割过程是先由工人在皮革上画出所有鞋样,然后再沿着画出的曲线手动切割。这种生产方法的缺点是:

1. 对每一个鞋样,不论其样式、大小如何,都需先画一遍,再切割一遍,不能灵活变通,因此切割速度缓慢、生产效率低下。例如同一样式、大小的皮鞋需生产 1000 双时,则其各个部分对应的鞋样各需画 1000 遍,再切割 1000 遍,人力资源耗费非常大。

2. 用人工直接在皮革上进行排版,工作效率低且难以实现各种鞋样的最优组合排列,皮革材料浪费严重。

3. 手动切割,不仅速度慢,还容易损坏皮革,浪费资源。

因此,为了提高皮革鞋样切割的效率,我们研制了鞋样设计和扩缩 CAD 系统。该系统将计算机技术与皮革切割的生产工艺相结合,具有广泛的实用价值。其主要由以下两大模块组成:

一、模板制作模块:

将放有各种标准鞋样的背景图片通过数码相机输入到计算机内,经图像分割、曲线拟合后得到各种鞋样边缘的矢量数据并保存在数据库中,整个过程示意图如下:

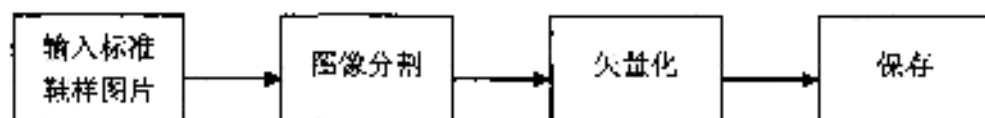


图 2.1 模板制作模块流程

二、自动切割模块:

用数码相机读入放在切割机上的皮革图片,从数据库中调入对应鞋样的矢量

数据并重画在屏幕上,在皮革图片上对这些鞋样进行缩放和重新排版,最后由计算机驱动切割机实现对重排后皮革鞋样进行自动切割。过程示意图如下:



图 2.2 自动切割模块流程

在整个处理过程中,鞋样的排版和大小变换相比原来的手工过程要灵活的多,这样就大大提高了工作效率,节省人力消耗,实现了生产的自动化和现代化并节约了皮革资源。

2.2 矢量化与鞋样系统

鞋样系统中为什么要进行矢量化。从前面的分析可知对模板图像进行分割以后可以得到鞋样的边缘点序列,能否直接将这些点序列存入数据库中,重排后直接进行切割。答案是不能。主要原因是以下四个:

1. 存储量过大。

提取出来的鞋样轮廓需要在计算机中长期保存,一个普通的鞋样边缘通常由几千个点组成,若直接保存这些点,需要大量的存储空间。

2. 离散的点序列不利于进行各种操作。

对离散数字曲线进行缩放、旋转等编辑需要变动曲线中每个点的坐标,因此编程实现起来非常繁琐且速度慢。而对矢量图形进行上述操作则非常方便。由于我们在后续的工作中经常需要对鞋样进行旋转、移动和缩放等,因此需要把点序列转换成矢量图形。

3. 边缘中存在细小的毛刺。

由于光照的影响及图片本身的特点,从模板图像中得到的初始鞋样边缘总是存在大量细小的毛刺。如图 2.3 是一幅只有一个鞋样的图片,我们将其中方框中的部分进行局部放大,看到的图片如图 2.4 所示:

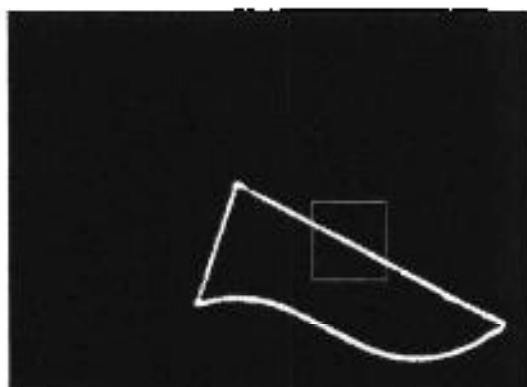


图 2.3 二值化后的鞋样图片



图 2.4 方框部分局部放大

这些细小的毛刺会大大降低切割的速度，并且容易使切割机产生抖动，损害皮革，因此在切割以前需要把离散边缘点序列转换成光滑的矢量曲线。

4. 切割机只能切割矢量图形。

由于以上四个原因，我们可以看出，在对鞋样进行切割之前将其进行矢量转换是非常重要的，矢量化是该系统中一个非常关键的工作。

第3章 图像分割

在对图像的研究和应用中,人们往往仅对各幅图像中的某些部分感兴趣,而对其它部分不感兴趣,这些人们感兴趣的部分常被称为目标或前景(其他部分称为背景),它们一般对应图像中特定的、具有独特性质的区域,如在皮革项目中图像的目标就是鞋样的边缘曲线,其它部分都为背景,目标相对背景有特定的灰度值范围。图像分割的目的就是将图像中人们感兴趣的部分从背景中分离出来,以进行后续的处理^[30]。分割的程度取决于要解决的问题,就是说,在应用中,当感兴趣的对象已经被分离出来时,就停止分割。例如在电子元件的自动检测方面,我们关注的是分析产品的图像,客观地检测是否存在特定的异常状态,比如缺失的元件或断裂的连接线路。超过识别这些元件所需的分割是没有意义的。在皮革项目中,我们最终要得到的是鞋样的数字化边缘曲线,因此,可以首先通过门限处理将鞋样区域分离出来,然后进行边缘跟踪得到鞋样的外边缘。

3.1 门限处理

图像分割算法一般是基于亮度值的两个基本特性之一:不连续性和相似性。不连续性的应用途径是基于亮度的不连续变化分割图像,比如图像的边缘。相似性的主要应用途径是依据事先制定的准则将图像分割为相似的区域。门限处理就属于这一类。

门限处理是一种人们普遍关注的用于分割处理的基础性方法,特别是在速度因素占重要地位的应用中。该方法由于其直观性和易于实现性,因此在图像分割的应用中处于中心地位。其基本原理如下:假设一幅图像由暗的对象和亮的背景组成,这样的组成方式将对象和背景具有灰度级的像素分成两组不同的支配模式。从背景中提取对象的一种显然的方法是选择一个门限值 T ,将这些模式分离开,然后所有 $f(x,y) < T$ 的点称为对象点;否则,就称为背景点。由于在鞋样照片中,我们是将黑色的鞋样放在白色的背景上,再加上灯光的影响,使得拍出来的数码照片中目标和背景的灰度值相差较大,如图3.1,因此采用单一的全局门限 T 就基本上能将需要的目标分离出来。

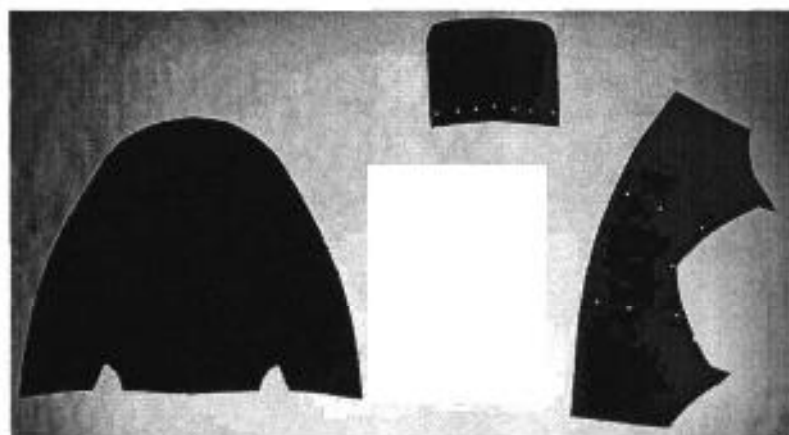


图 3.1 鞋样图片

该门限 T 通过一种自动迭代的方法进行选取, 选取步骤如下^[11]:

1. 选择一个 T 的初始估计值。
2. 用 T 分割图像。这样做会生成两组像素: $G1$ 由所有灰度值大于 T 的像素组成, 而 $G2$ 由所有灰度值小于或等于 T 的像素组成。
3. 对区域 $G1$ 和 $G2$ 中的所有像素值计算平均灰度值 u_1 和 u_2 。
4. 计算新的门限值:

$$T = \frac{1}{2}(u_1 + u_2)$$

5. 重复步骤 2 到 4, 直到逐次迭代所得的 T 值之差小于事先定义的参数 T_0 。

当有理由认为背景和对象在图像中占据的面积相近时, 则好的 T 初始值就是图像中的平均灰度值。如果对象所占面积小于背景 (反之亦然), 则一个像素组会在直方图中占主要地位, 平均灰度值就不是好的初始选择。此时, T 更合适的初值是诸如灰度值的中间值一类的初值。参数 T_0 用于在参数 T 变得很小时停止算法的运行。这个参数在迭代的速度问题比较重要的时候应用。

图 3.2 显示了使用基于前述算法估计的全局门限对图像 3.1 进行分割后得到的结果, 其中阈值 T 的初值使用的是平均灰度级, 同时令 $T_0=0$, 进行三次迭代后得到最终的阈值。

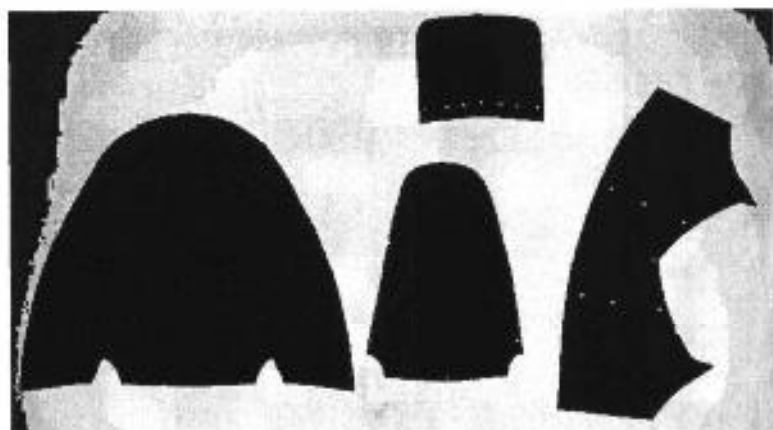


图 3.2 图 3.1 进行阈值分割后的结果

3.2 噪声去除

由图 3.2 可以看到一部分背景也被当作目标分割出来了，且图片中还有一些小噪声。这是由于图片自身的反光以及采集过程中噪声的影响，使得任何一种分割方法都难以保证将目标百分之百准确的分割出来。而这些误分割和噪声又会影响后续的工作，因此需采用手动方法对分割过程中产生的阴影或噪声进行消除。图 3.3 是手动操作后得到的图像。

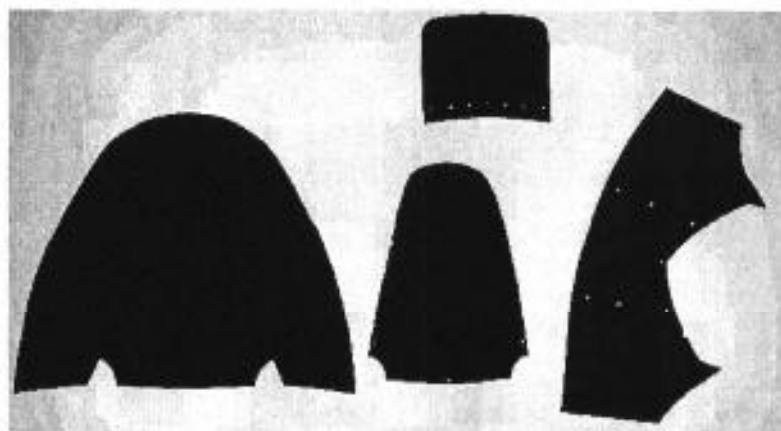


图 3.3 手动去噪后的鞋样图片

3.3 边缘跟踪

目标分离出来以后，就可以进行边缘跟踪。在本小节中我们介绍用 8 方向 freeman 链码对图像中的目标部分进行边缘跟踪。链码是对边界点的一种编码表示方法，其特点是利用一系列具有特定长度和方向的相连的直线段来表示目标的边界。因为每个线段的长度固定而且方向数目有限，所以只有边界的起点需要用（绝对）坐标表示，其余点都可只用接续方向来代表偏移量，最后通过起点的绝

对坐标与各个偏移量计算出边缘中各点的绝对坐标。光栅图像一般是按固定间距的网格采集的,所以最简单的链码是跟踪边界并赋给每2个相邻像素的连线1个方向值。最常见的是4方向和8方向链码。它们的共同特点是线段的长度固定,方向数有限。

3.3.1 像素的邻接性

对一个坐标为 (x, y) 的像素 p , 它可以有四个水平和垂直的邻近像素。它们的坐标分别为 $(x+1, y)$, $(x-1, y)$, $(x, y+1)$, $(x, y-1)$ 。这些像素 (用 r 表示) 组成 p 的4-邻域, 记为 $N_4(p)$ 。坐标为 (x, y) 的像素 p 与它各个4-邻域近邻像素是1个单位距离。像素 p 的4个对角近邻像素 (用 s 表示) 的坐标是 $(x+1, y+1)$, $(x+1, y-1)$, $(x-1, y+1)$, $(x-1, y-1)$, 它们记为 $N_D(p)$ 。这些像素点加上 p 的4-邻域像素合称为 p 的8-邻域。

3.3.2 像素的连通性

像素间的连通性是一个基本概念。它简化了许多光栅图像概念的定义, 如区域和边界。为定义像素间的连通性, 定义以下三种邻接情况:

(1) 4-邻接: 2个像素 p 和 r 在 V 中取值且 r 在 $N_4(p)$ 中, 则它们为4-邻接。

(2) 8-邻接: 2个像素 p 和 r 在 V 中取值且 r 在 $N_8(p)$ 中, 则它们为8-邻接。

(3) m -邻接 (混合连接): 2个像素 p 和 r 在 V 中取值且满足下列条件之一, 则它们为 m -连接。

判断两个像素是否相连通, 必须确定它们是否满足以上三种邻接情况之一及它们的灰度值是否满足特定的相似性准则 (或者说, 它们的灰度值是否相等)。例如, 在具有0、1的二值图像中, 两个像素可能是4-邻接的, 但是, 仅仅当它们具有同一灰度值时才能说是连通的。

对于一个八连通的图像区域来说, 由以上定义可知, 区域中的每个像素, 它的周围总存在8个像素与它连接, 可以为这八个像素设定从0到7的方向编号, 如图3.4所示。这样一旦确定了像素 p 的位置, 以及某个邻接像素的编码, 就可

以知道邻接像素的位置。链码实质上是一串指向符的序列。链码表示就是从某点开始观察某一曲线的走向并用相应的指向符来表示,结果形成一个序列,因此可以用链码来描述任意曲线或闭合的边界。

3	2	1
4	P	0
5	6	7

图 3.4 链码的 8 邻域方向

本文中用链码对图像中的目标进行跟踪的过程如下^[30]:对于整个图像进行从上到下、从左到右的扫描,如果发现了一个作为前景颜色的像素(黑点),则对和这一点相关的图形进行外边缘的跟踪,跟踪完毕后擦除掉跟踪得到的外边缘内部的所有黑像素,再对图像进行扫描跟踪,直到整个图像中已经不再有黑色像素。

对于每个特定目标进行外边缘跟踪的过程如下:从第一个边界点开始,定义起始点的初始搜索方向为左上方;如果左上方的点是黑点,则为边界点,否则搜索方向顺时针旋转 45 度。直到找到第一个黑点为止,这个黑点就是新的边界点。新边界点的初始搜索方向为前一边界点到该点方向的反方向并顺时针旋转 45 度,用同样的方法搜索下一点,直到返回最初的边界点为止。其算法实现如下:其中: n 表示当前方向数, $CurPoint$ 表示当前点, $NewPoint$ 表示在当前点的 n 方向上的点, $StartPoint$ 表示起始点

```
do
{
     $n = (n + 4) \% 8;$  //回到起始方向
}
do
{
     $n = (n + 7) \% 8;$  // turn around 顺时针旋转45度
     $NewPoint = CurPoint$  点在  $n$  方向上的点;
} while ( $NewPoint$  不为目标点)
if ( $NewPoint == StartPoint$ )
{
    本次跟踪结束;
```

```
        ;  
    else  
    {  
        保存当前方向n;  
    }  
    CutPoint = NewPoint;  
}while(1)
```

对图片中的每一个目标都进行一次链码跟踪,并记录下每段链码中起始点的坐标值,然后根据起始坐标及当前列值将链码序列转换成边缘的坐标点序列。

第4章 边缘曲线的分段拟合

4.1 基本概念

图像分割以后,鞋样的边缘轮廓被表示成一个个分散的像素点,数据量非常大,离需要的图元对象还差很远,不利于进行存储和图形编辑。因此,通常的处理方式是用一系列首尾相连的曲线来拟合这些点序列,以达到最终矢量化的目的。轮廓曲线通常都采用内插曲线或者逼近曲线来实现拟合。在研究和应用曲线拟合时,对插值、逼近和拟合这些术语的含义要了解清楚。

4.1.1 插值、逼近和拟合

1. 插值

插值是函数逼近的重要方法。给定函数 $f(x,y)$ 在区间 $[a,b]$ 中互异的 n 个点的值 $f(x_i)(i=1,2,\dots,n)$, 基于这个列表数据, 寻找某一函数 $\varphi(x)$ 去逼近 $f(x)$ 。若要求逼近函数 $\varphi(x)$ 在 x_i 与 $f(x_i)$ 相等, 就称这样的函数逼近问题为插值问题, 称 $\varphi(x)$ 为 $f(x)$ 的插值函数。也就是说, $\varphi(x)$ 在 n 个插值节点 x_i 处与 $f(x_i)$ 相等, 而在别处就用 $\varphi(x)$ 近似代替 $f(x)$ 。在曲线中最常用的是线性插值和抛物样条插值。

2. 逼近

上面讨论了已知型值点(插值点)的插值方法。当型值点太多时,构造插值函数使其通过所有的型值点相当困难。客观上看,由于过多的型值点也会有误差,也没必要寻找一个插值函数通过所有的型值点。这时,往往选择一个次数较低的函数,在某种意义上最佳逼近这些型值点。逼近的方法有很多,最常用的方法是最小二乘法。假设已知一组型值点 $(x_i, y_i)(i=1,2,\dots,n)$, 要求构造一个 $m(m < n-1)$ 次多项式 $y = f(x)$ 逼近这些型值点。逼近的好坏主要看是否使各点偏差的平方和最小,即:

$$\varphi = \sum_{k=1}^n [F(x_k) - y_k]^2 \quad (k=1,2,\dots,n) \quad (4.1)$$

最小。

令 $F(x)$ 为一个 m 次多项式:

$$F(x) = \sum_{j=0}^m a_j x^j \quad (4.2)$$

这里的最小二乘问题就是要给出系数 a_j , 使残差平方和达到极小值:

$$\varphi(a_j) = \sum_{k=0}^n \left[\sum_{j=0}^m a_j x_k^j - y_k \right]^2 \quad (4.3)$$

根据一般求极值的方法可知, 使式(4.3)达到极小的 $a_j (j=0, 1, \dots, m)$, 必须满足式所示的方程组:

$$\frac{\partial \varphi}{\partial a_j} = 2 \sum_{k=0}^n \left[\sum_{j=0}^m a_j x_k^j - y_k \right] x_k^j = 0 \quad (4.4)$$

共有 $m+1$ 个方程, 可解出 $m+1$ 个未知数 a_0, a_1, \dots, a_m , 代入定义即可求出多项式函数 $F(x)$ 逼近已知的 n 个型值点。

3. 拟合

拟合并不像上述的插值、逼近那样有完整的定义和数学表达。拟合是指在曲线、曲面的设计过程中, 用插值或者逼近的方法使生成的曲线、曲面达到某些设计要求, 如在允许的范围内贴近原始的型值点或控制点序列等。在本文中采用的是抛物样条插值的方法拟合原始的点序列, 目的是使鞋样边缘更光滑自然, 便于编辑和切割。

4.1.2 误差计算

在讨论曲线拟合时, 首先应保证拟合的误差足够小, 才能切割出准确的鞋样。目前有很多种计算拟合曲线与候选边缘点误差的方法, 以下是一些常用的方法, 其中 d_i 是边缘点到一条拟合曲线的距离:

1. 最大绝对误差 (maximum absolute error, MAE)

测量最坏情况下边缘点偏离拟合曲线的距离, 其计算如式(4.5)

$$MAE = \max |d_i| \quad (4.5)$$

2. 均方差 (mean squared error, MSE)

给出边缘点偏离拟合曲线的总的测度, 如式(4.6)

$$MSE = \frac{1}{n} \sum_{i=1}^n d_i^2 \quad (4.6)$$

3. 规范化最大误差 (normalized maximum error, NME)

最大绝对误差与拟合曲线长度 S 之比, 如式(4.7)

$$\epsilon = \frac{\max |d_i|}{S} \quad (4.7)$$

在鞋样 CAD 系统中, 允许的最大实际误差是 2 毫米, 根据所采用的切割机与屏幕大小的比例, 实际中的 2 毫米大约对应屏幕上的 5 个像素点。本文中采用最大绝对误差衡量拟合算法的精度, 在计算机中, 当原始边缘点与拟合曲线之间的最大距离不超过 5 个像素时, 认为算法满足精度要求。

4.1.3 算法的评价标准

在 CAD 系统里, 评价一个拟合算法效果好坏的标准主要是以下两个方面:

1. 精确性: 拟合后的曲线段应与原始的边缘点序列误差比较小, 并能表示出原始轮廓的特征, 如在本系统中要求拟合曲线与原始点序列的最大误差不能超过 5 个像素, 且要突出轮廓在尖点处的特征。

2. 数据压缩率: 曲线拟合应当提高轮廓表示的经济性, 能为图像的后处理提供一种更简单、更紧凑的表示。如边缘轮廓最容易实现的表示形式是通过链码直接得到的边缘点序列, 这种表示具有很高的精确度, 但其表示的紧凑性是最差的, 因此不是一种有效的图像分析表示方式。又如一条直线上的边缘点集用一条直线来拟合是表示这些边缘点的最简单和最有效的方法, 这一表示也简化了后处理 (如确定线的长度和走向)。

曲线拟合的高效性及精确性主要由以下两个因素决定: 待拟合曲线形状的特点和拟合算法本身的性能。在不同的 CAD 系统里处理的图片的形状都具有其自身的特点。因此要根据系统特定的要求来选取不同的拟合算法。在本系统里, 一个数字化的鞋样边缘除了数据量大和存在毛刺以外, 还具有以下三个特征: (见图 2.2、2.3)

(1) 存在曲率变化较缓慢的曲线段;

(2) 存在尖点;

(3) 鞋样边缘是闭合的;

综合以上特点, 本文提出了一种分段拟合的方法对鞋样边缘进行矢量化。该方法主要分为两大步骤:

步骤一: 尖点提取, 即通过曲率方法提取出轮廓中的尖点, 防止在下一步的拟合中将尖点平滑掉, 曲线特征丢失。

步骤二: 曲线拟合, 即以尖点为分段点, 对每两个尖点之间的曲线段采用样条曲线进行递归拟合, 直到精度满足一定条件为止。

以下两节对这两个步骤分别进行详细介绍。

4.2 尖点提取

尖点对应曲线中具有局部曲率极大值的点, 即曲线的斜率变化发生突变的点。如图 4.1 中的以 ‘•’ 标示的点即为一个尖点。尖点一般位于两段光滑曲线的分段处, 在拟合时无法用光滑的曲线表示出来, 因此通常先提取出边缘中的尖点, 然后把两个尖点之间的曲线作为一个拟合段。

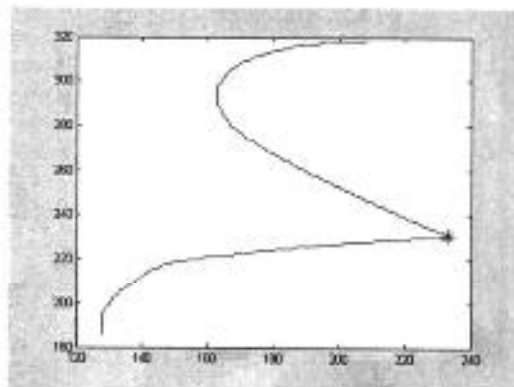


图 4.1 尖点示意图

现有的提取尖点的算法大多利用曲率局部极大值的方法, 因此曲率的计算是关键, 由于在数字曲线中, 无法直接用公式准确的计算出某一点的曲率值, 因而, 只能计算曲率的某种近似值, 一般来说在数字曲线中估算曲率主要有以下两种方法: 余弦法和弦长比法。

(1) 余弦法^[32]

余弦法采用偏角余弦值来估算曲率。设 p_i 点的曲率为 c_i , 这里 p_i 点为曲

线上的第 i 点, 定义 p_i 点的 k 矢量:

$$\bar{a}_k = (x_i - x_{i+k}, y_i - y_{i+k}) \quad (4.8)$$

$$\bar{b}_k = (x_i - x_{i-k}, y_i - y_{i-k}) \quad (4.9)$$

则:

$$c_i = \cos \theta_i = \bar{a}_k \bullet \bar{b}_k / (|\bar{a}_k| \bullet |\bar{b}_k|) \quad (4.10)$$

这里, θ_i 为 k 矢量 \bar{a}_k 和 \bar{b}_k 之间的夹角, 式(4.10)得到的结果 c_i 即为曲线在该点的余弦值。余弦值越大的点曲率也越大。

(2)弦长比法^[33]

数字曲线中连接点 p_{i-k} 和点 p_{i+k} 的直线方程为

$$Ax + By + C = 0 \quad (4.11)$$

其中

$$A = (y_{i-k} - y_{i+k}) \quad (4.12)$$

$$B = (x_{i-k} - x_{i+k}) \quad (4.13)$$

$$C = x_{i-k}y_{i+k} - x_{i+k}y_{i-k} \quad (4.14)$$

而 $L_k = \sqrt{A^2 + B^2}$ 是点 (x_{i-k}, y_{i-k}) 和点 (x_{i+k}, y_{i+k}) 之间的弦长。点 p_i 到该弦的距离为:

$$d_k = \frac{|Ax_i + By_i + C|}{L_k} \quad (4.15)$$

则 p_i 点的曲率 c_k 可以表示为:

$$c_k = d_k / L_k \quad (4.16)$$

显然, 由该式计算得到的 c_k 值越大, 说明该点的曲率越大, 这点也越有可能是尖点。在上述两种方法中均用到常数 k , k 表示曲率近似的范围, 一般可取 $k=5 \sim 15$ 。

其他还有一些别的计算曲率的方法, 如利用 B 样条曲线进行拟合获得曲率

[34]; 利用某点前后几点的几何中心与该点所构成的向量计算曲率角^[15]; 以及利用轮廓内面积与轮廓外面积在以轮廓点为圆心的圆中所占百分比估计曲率等等^[17]。以上方法都是直接利用曲线上点的坐标来计算曲率, 计算比较复杂。本文针对鞋样边缘闭合的特点, 提出了一种间接的算法, 该算法相对以上提到的各种算法计算量要小、算法简单, 并且适合任意形状的曲线, 能快速准确的提取出复杂轮廓上的尖点。

4.2.1 算法思想

设 $P(i)=[X(i), Y(i)](i=0, 1, 2 \cdots n-1)$ 为一闭合的图像边缘轮廓点序列 (见图 4.2), 用 “*” 标示的 $P(0)$ 点为该序列的起始点, 随着 i 的增大, $P(i)$ 沿逆时针方向行进一周后又回到 $P(0)$ 点。由于在闭合曲线上, X 和 Y 之间是非函数关系, 因此, 函数中曲率的计算公式不能直接用在此处。

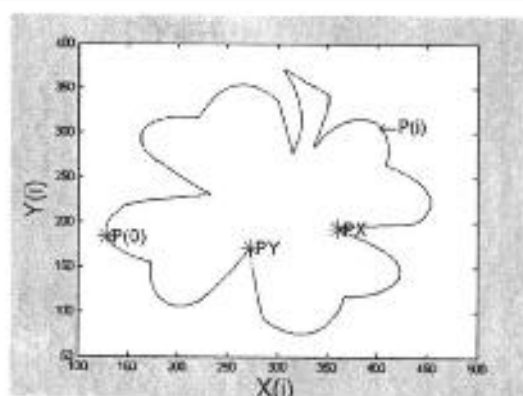
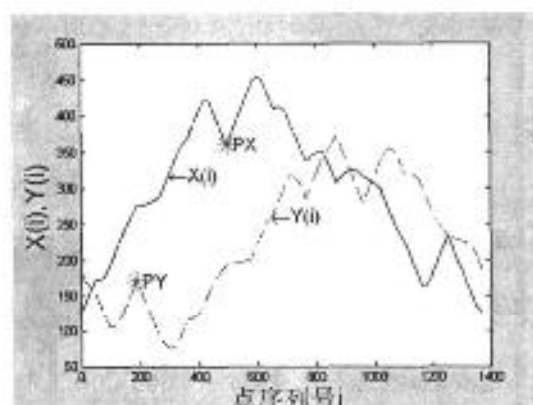


图 4.2 闭合图形轮廓点序列 $P(i)$

在图 4.2 上可以看到, $P(i)$ 点的坐标由 $X(i)$ 和 $Y(i)$ 共同决定, 因此, 可将 $P(i)$ 分解为两条一维离散曲线 $X(i), Y(i)$, (如图 4.3), $X(i)$ 随 i 的变化过程即为 $P(i)$ 在水平方向上的变化, $Y(i)$ 则为 $P(i)$ 在竖直方向上的变化, 因此, 只要分别找出曲线 $X(i), Y(i)$ 上的尖点, 综合起来, 就能得到 $P(i)$ 上的尖点集合。例如在图 4.2、图 4.3 中, 由点坐标的变化趋势可以看出, 曲线 $X(i)$ 上的尖点 PX 和曲线 $Y(i)$ 上的尖点 PY (“*” 标示), 分别对应曲线 $P(i)$ 上的尖点 PX 和 PY (“*” 标示)。由于 $X(i), Y(i)$ 都是关于 i 的一元离散函数, 因此曲率的估算简单明了。

图 4.3 $P(i)$ 的分量 $X(i), Y(i)$

4.2.2 算法数学基础

如何计算 $X(i)$ 和 $Y(i)$ 上各点的曲率也是本算法中一个重要部分。由于我们在检测尖点时是根据曲率绝对值的大小进行的（即曲线斜率变化绝对值的大小），因此并不需要知道曲率本身的准确值，只要能根据某一数值判断其绝对值大小以对曲线上的点进行取舍即可，我们设该数值为 q ，为简化运算，在本算法中取 q 等于近似曲率绝对值的整数倍。下面以 $X(i)$ 为例说明如何计算 q 。在数值分析中^[7]，对于离散的函数 $X = X(i)$ ，有：

表 4.1 离散函数 $X(i)$

i	0	1	2	...	$n-1$
X	$X(0)$	$X(1)$	$X(2)$...	$X(n-1)$

$i=0,1,2,\dots,n-1$ 为等距节点，由插值原理可得：

$$X''(i) \approx \frac{1}{h^2} [X(i-h) - 2X(i) + X(i+h)] \quad (4.17)$$

又曲线上点的二阶导数近似等于该点的曲率^[35]，设 $X(i)$ 上第 i 点的曲率为 K ，则有：

$$K \approx \frac{1}{h^2} [X(i-h) - 2X(i) + X(i+h)] \quad (4.18)$$

考虑到数字化及噪声的影响，在本算法中取步长 $h=10$ ，且令 $q = |h^2 \times K|$

$$\text{则} \quad q = |X(i-10) - 2X(i) + X(i+10)| \quad (4.19)$$

进一步变形得

$$q = |X(i+10) - X(i) - X(i) + X(i-10)| \quad (4.20)$$

同理, $Y(i)$ 中各点的 q 值计算方法与 $X(i)$ 中一样, 在此不再重复。

4.2.3 算法设计

根据以上的曲率近似值即 q 值求曲线尖点的算法思想如下: 以相邻且其 q 值全大于 0 的点序列作为一个局部曲线段, 找出该段曲线中 q 值最大的点, 若该点的 q 值大于某一阈值 T , 则将该点标记为一个尖点, 否则则为非尖点。具体设计如下, 其中 max 用来记录当前段中的最大 q 值, k 用来记录当前段中具有最大 q 值的点的序列号, T 为阈值:

(1) 求曲线 $X(i)$ 上的尖点

计算 $X(i)$ 上某一点对应的 $q(i)$ 值并进行判断:

```

if  $q(i) > 0$ 
{
    if  $q(i) \geq max$ 
    { $max = q(i)$ ;  $k = i$ ; 计算  $q(i+1)$  并重新开始判断; }
    else
    {计算  $q(i+1)$  并重新开始判断; }
}
else
{
    if  $max > T$ 
    {将第  $k$  点标记为尖点;  $max = 0$ ; 计算  $q(i+1)$  并重新开始判断; }
    else
    { $max = 0$ ; 计算  $q(i+1)$  并重新开始判断; }
}

```

(2) 求曲线 $Y(i)$ 上的尖点

具体过程与求 $X(i)$ 上的尖点相同。

(3) 综合 $X(i)$ 与 $Y(i)$ 上的尖点, 求原曲线 $P(i)$ 上的尖点集。

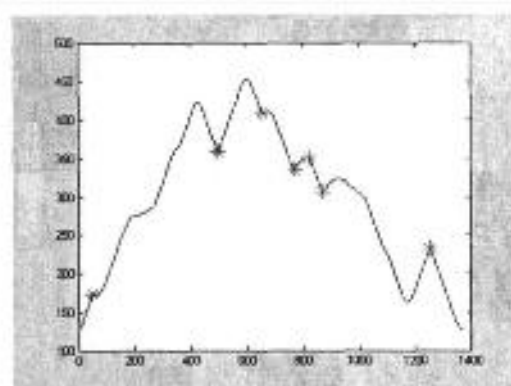
对于 $X(i)$ 上的任意特征点 P_x 和 $Y(i)$ 上的任意尖点 P_y , 设其点的序列号分别为 j, k ,

若 $|j - k| \leq 3$, 则 P_x 和 P_y 对应 $P(i)$ 上的同一个尖点。

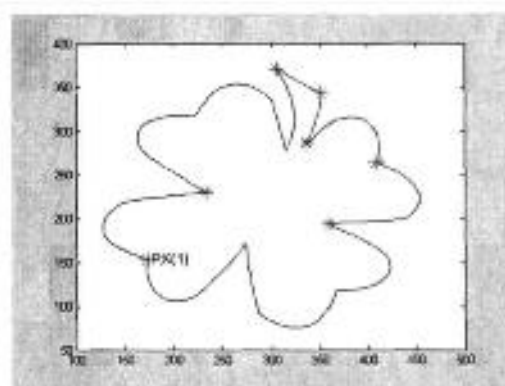
若 $|j - k| > 3$, 则 P_x 和 P_y 对应 $P(i)$ 上两个不同的尖点。

4.2.4 实验结果及分析

以下几个图是用 MATLAB^[36] 语言实现该算法并对图 4.2 中的边缘提取尖点得到的结果, (图中 ‘*’ 代表尖点)。图 4.4 (a) 是在曲线 $X(i)$ 上提取的尖点的情况, 图 4.4 (b) 是 $X(i)$ 中的尖点在原图中的对应, 其中原图中的 $P_x(1)$ 点对于 $X(i)$ 中的第一个尖点, 其后的以逆时针方向与 $X(i)$ 中的尖点一一对应。图 4.5 (a) 与图 4.5 (b) 则是 $Y(i)$ 中的尖点提取及其在原图中的对应情况。

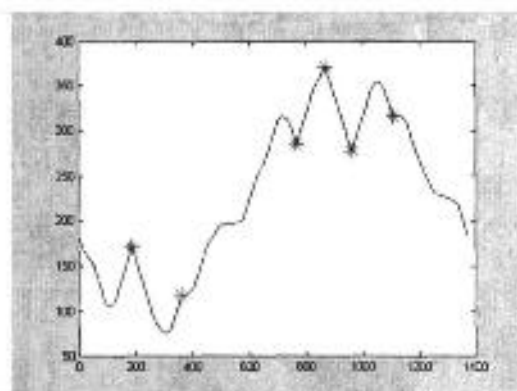


(a)

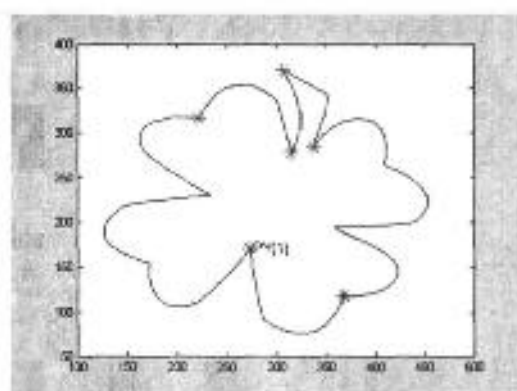


(b)

图 4.4 曲线 $X(i)$ 上的尖点及其在原图中的对应



(a)



(b)

图 4.5 曲线 $Y(i)$ 上的尖点及其在原图中的对应

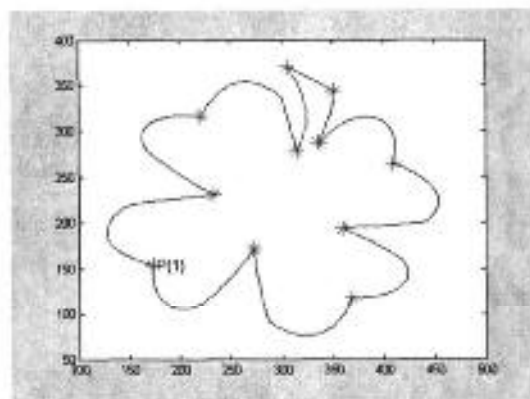


图 4.6 综合两个方向上的尖点后的结果

图 4.6 则是用上面描述算法中的第三步对两个方向上的尖点进行尖综合得到的结果。

同样我们又用该算法对其他很多鞋样边缘进行尖点提取得到的一些结果如图 4.7 所示:

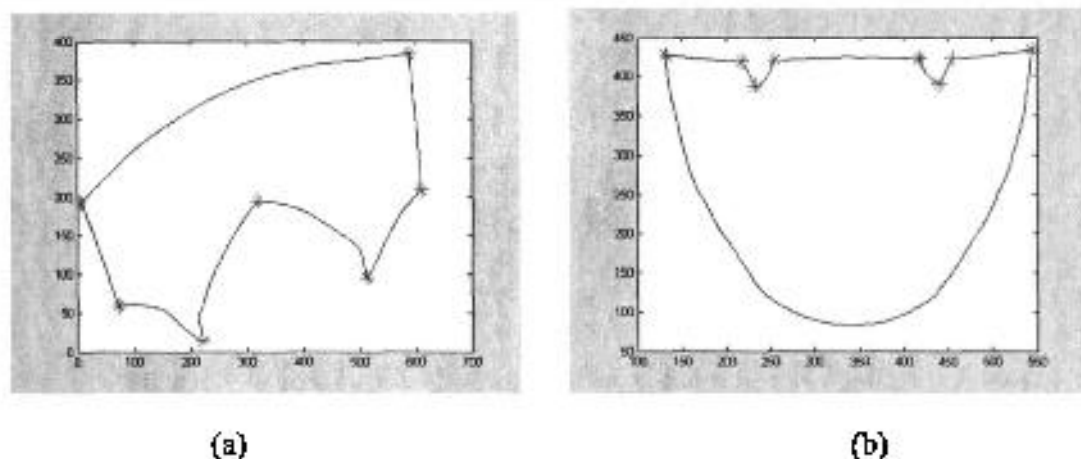


图 4.7 其他鞋样

从以上结果可以看出该算法对尖点的定位是比较准确的,且适合各种形状。在算法效率上,本算法在两个方向上点的曲率时,分别只用到了三次加减法运算(见式 4.20),与其他方法比较,计算量大大减小,效率高,以下是该算法在计算某点的曲率时其计算量与余弦法和弦长比法的一个比较:

表 4.2 各种曲率计算方法计算量的比较

方法 \ 操作	加减	乘除法	开方运算
余弦法	7	7	2
弦长比法	6	7	0
本方法	6	0	0

本节中提出的间接方法实际上是一种基于参数方程的方法,在该方法中我们引入了一个新的参数,即点序列号 i ,利用该参数将点序列 $P(i)$ 的坐标变为参数方程 $X(i)$, $Y(i)$ 的方法来表示

$$\begin{cases} X(i) = X_i \\ Y(i) = Y_i \end{cases} \quad i = 0, 1, 2, \dots, n \quad (4.21)$$

其中 $X(i)$, $Y(i)$ 是关于 i 的一维离散函数,因此可以对其用函数的方法来处理。

目前很多数字曲线的处理方法中,都要求坐标 y 是坐标 x 的函数,如最小二乘拟合法。因此这些方法无法处理闭合形状的曲线,如本项目中的鞋样曲线。但在引入了参数方程以后,我们可以把一条闭合曲线表示成两条离散函数曲线,因此就可以把这些方法应用到闭合曲线的处理上了。因此,本节中提出的间接方法不光可以用在曲率计算中,对其他很多数字曲线的处理方法也有意义。鉴于时间和篇幅的关系,不再对此方法进行详细的扩展分析。

4.3 曲线拟合

4.3.1 抛物样条曲线

矢量化中用来拟合的曲线通常包括线段、样条曲线、圆弧及椭圆弧等,它们在一般的 CAD 系统里都有对应的表达形式。在本系统里,由于鞋样边缘的变化多样性,因此本文中选择形状比较灵活的抛物样条曲线作为拟合基元。通过不在同一直线上的三点: P_1, P_2, P_3 , 定义一条抛物样条曲线的表达形式如下:

$$P(t) = A_1 + A_2 t + A_3 t^2 \quad 0 \leq t \leq 1 \quad (4.22)$$

该抛物样条曲线过 P_1, P_2, P_3 三个点,并且有:

- (1) 抛物线段以 P_1 为起始点。即当参变量 $t=0$ 时,曲线过 P_1 点。
- (2) 抛物线段以 P_3 点为终点,即当参变量 $t=1$ 时,曲线过 P_3 点。
- (3) 当参变量 $t=0.5$ 时,曲线过 P_2 点,且切矢量等于 $P_3 - P_1$ 。

由以上三个条件得到方程组如下:

$$\begin{cases} A_1 = P_1 \\ A_1 + A_2 + A_3 = P_3 \\ A_1 + 0.5A_2 + 0.25A_3 = P_2 \end{cases} \quad (4.23)$$

解得:

$$\begin{cases} A_1 = P_1 \\ A_2 = 4P_2 - P_3 - 3P_1 \\ A_3 = 2P_1 + 2P_3 - 4P_2 \end{cases} \quad (4.24)$$

将式(4.24)代入式(4.22)得:

$$P(t) = (2t^3 - 3t + 1)P_1 + (4t - 4t^2)P_2 + (2t^2 - t)P_3 \quad 0 \leq t \leq 1 \quad (4.25)$$

由式(4.24)可得, 当 P_1, P_2, P_3 在同一条直线上时, 即 $2P_2 = P_1 + P_3$ 时, $A_3 = 0$, 曲线方程变成一条直线。因此在本文中的拟合基元实际上是直线和抛物样条曲线。 $P(t)$ 是一个点向量, 在二维平面上它包含了两个坐标值 $[x(t), y(t)]$ 。以上推导求出的算式, 即为我们所要求的过不在一直线上的三点: $P_1(x_1, y_1), P_2(x_2, y_2)$ 和 $P_3(x_3, y_3)$ 的抛物线方程。对该条抛物线进行存储和编辑时, 只需对 P_1, P_2 和 P_3 进行操作。无需像数字曲线那样对每一个点都进行操作。在屏幕上显示该条曲线时, 可以根据参变量 t 的取值, 一一计算出位于曲线上的数据点, 然后顺次连线绘出图形。

4.3.2 现有拟合算法

由于鞋样轮廓的变化多样性和不确定性, 即使是两个尖点之间的点序列, 也很难用单一的一条曲线将其准确的拟合, 通常需再分成很多小段来进行拟合。如何正确选取分界点, 使分段拟合时即能满足精度的要求, 又能大大减少点的数量, 这是曲线拟合中的一个关键问题。由于原始的点序列之间没有明显的规律性, 因此难以通过其自身的特点找出最佳的分界点序列, 以实现最优的拟合。所谓的最优算法, 如遗传算法等, 也只是通过在不同解之间进行不断迭代, 直到找到满足条件的解为止。因此这种算法的计算量是非常大的, 不适合长的曲线段。

目前的曲线拟合算法中用的最多的是遗传算法^[24, 25, 26, 29], 遗传算法拟合曲线的思想如下: 把待拟合的点序列表示为一段二进制编码 $S = s_1s_2 \cdots s_1 \cdots s_n$, 其中 s_i

的取值为 0 或 1, 当 s_i 等于 1, 则表示点序列中的第 i 点为一个分界点。S 的每一种取值称为一个染色体, 遗传算法通过选择、交叉及变异等操作在各染色体之间进行迭代, 直到找出满足适应度条件的一条染色体作为问题的解, 通常适应度的计算公式如下:

$$F(C_i) = \frac{1}{Err(C_i) \times Sn^m} \quad (4.26)$$

式中 $Err(C_i)$ 为染色体中所有段的拟合误差之和, Sn 为曲线的段数, m 则为调整系数, m 的值可以根据需要进行修改。由上述分析可知遗传算法通过设定适应度值的大小实现较高精度的拟合和较好的数据压缩。

遗传算法中存在的主要问题是当待拟合的点序列数量非常大时, 染色体的取值范围也非常大, 使得迭代的次数增大, 且每次迭代中计算当前染色体的适应度时, 需要对整体的拟合误差进行计算, 计算量非常大。设一段曲线中待拟合点序列的个数为 n , 那么其表示的染色体有 2^n 种取值, 最坏情况下需要迭代 2^n 次才能找到需要的解。这种计算量对于工业 CAD 系统没有实际的应用价值, 因此人们提出了很多改进算法, 如文献[29]提出的基于预选关键点的遗传算法, 该算法首先根据曲率大小选出部分点作为候选分界点, 然后在这些候选分界点之间进行迭代, 从而大大减少了待选择点的数量, 减少了迭代次数。假设通过曲率计算得到的预选点的数量为 $m(< n)$, 则可以将迭代的次数减小到 2^m 次, 相对原算法计算量大大减小, 但对于较长的曲线, 当需要的分界点个数比较多时, 其计算量还是太大。

4.3.3 递归算法

由以上分析可知遗传算法虽然能满足精度和压缩率的要求, 但是算法复杂, 计算量太大。因此, 本文提出了一种递归的拟合方法对各个尖点之间的曲线段分别进行拟合。该算法思想如下: 以当前离散曲线段的起点、中点和末点构造一条抛物样条曲线对当前曲线段进行拟合并进行误差计算, 若本段拟合的误差小于给定的阈值, 则返回; 否则以当前曲线段的中点将曲线分为两部分再用同样的方法分别进行拟合, 算法设计如下, 其中:

first 表示当前曲线段的起始点。

last 表示当前曲线段的终点。

T 为误差阈值，单位为像素，可以根据不同需要进行设定

Recursive (int first, int last)

```
{  
    bew = first和last的中点;  
    构造依次过first, bew和last三点的抛物样条曲线;  
    Len = 当前段的拟合误差;  
    if (Len >= T)  
    {  
        Recursive(first, bew);  
        Recursive(bew, last);  
    }  
    else  
        {返回; }  
}
```

用这种方法依次拟合各尖点之间的曲线段，就能得到总的拟合结果。可以看出该算法对每段曲线只需用两个简单的递归语句就能实现拟合，相对于遗传算法中各种复杂的遗传、变异及交叉等操作，非常简单且易于实现。

4.3.4 算法分析及试验结果

由前面的分析可知，评价一个矢量化算法的根本标准是其精确性和其数据压缩率。以下就从这两个方面对递归算法进行分析：

1. 精确性

递归算法中递归的中止条件是某段曲线的拟合误差小于误差阈值 T，若条件不满足，则可以对曲线进行进一步的细分，以进行更精确的拟合，直到满足精度条件为止。由于 T 可以根据实际需要进行设定，因此理论上递归算法能达到任意大小的拟合精度。但精度越大，递归次数越多，计算量也越大。

2. 数据压缩率

拟合后的鞋样边缘可以表示为一系列首尾相连的抛物样条曲线。由前面的分析可知一条抛物样条曲线可以由不在同一直线上的三个点决定, 因此对它们进行存储和编辑时无需直接操作曲线, 只需存储或编辑各段曲线的分界点及其在原始曲线上的中点 (参见 4.3.1 及 4.3.3 节), 在最后显示或切割时再用这些分界点及中点作为参数恢复出各条曲线。因此本算法的压缩率就是原始点序列个数与分界点及中点个数的比。比率越大, 则算法的压缩效率越大。在递归算法中, 最后得到的分界点的数量也与误差阈值 T 的选取有关, T 越小, 拟合的精度越高, 则对原曲线的分段就越细, 分界点数量越多, 压缩率就越小。同样, T 越大, 则压缩率也越大。因此实际应用中我们应当根据需要选择合适的阈值, 这样可在满足精度的条件下减小递归次数, 避免不必要的计算, 同时还可以提高数据压缩率。

以下各组图片显示了用该算法对各种形状进行拟合, 在误差阈值 T 分别取 5 和 12 的时候得到的结果 (图中线条有分叉的地方表示存在一定的拟合误差, 无分叉的地方则表示拟合的精度较高, 圆圈表示各段抛物样条曲线间的分界点)。



图 4.8 原始形状 I

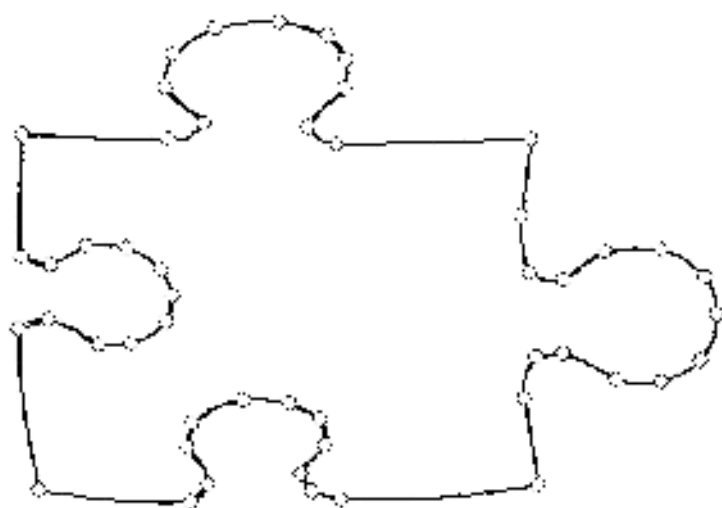


图 4.9 $T = 5$ 时的拟合情况及分界点

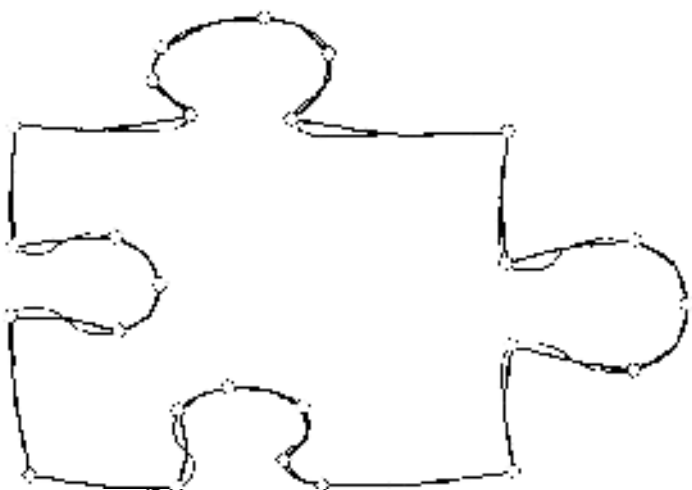


图 4.10 $T = 12$ 时的拟合情况及分界点



图 4.11 原始形状 2

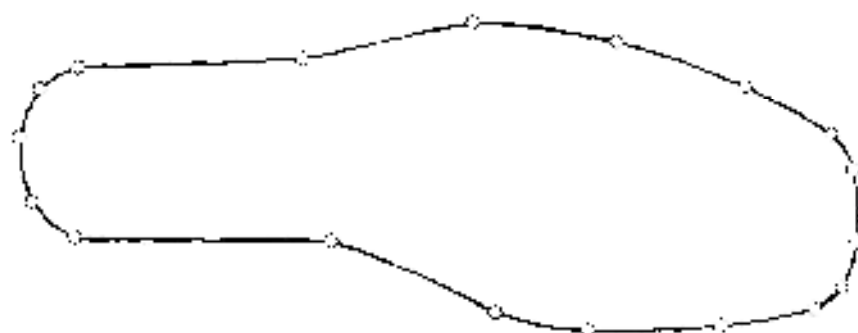
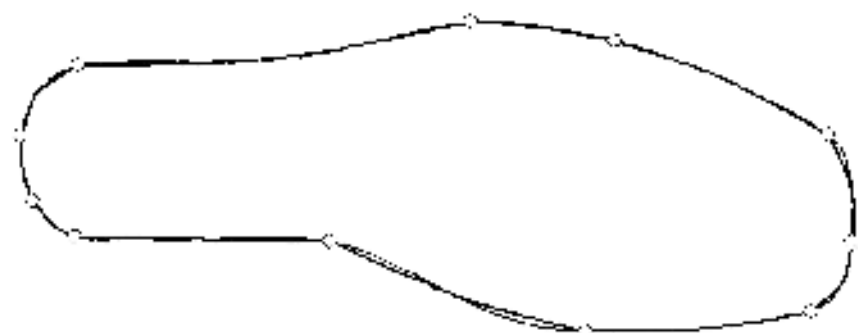
图 4.12 $T = 5$ 时的拟合情况及分界点图 4.13 $T = 12$ 时的拟合情况及分界点



图 4.14 原始形状 3

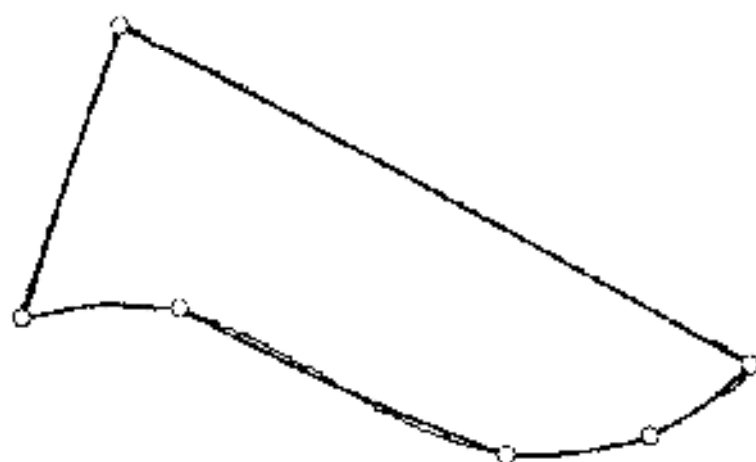
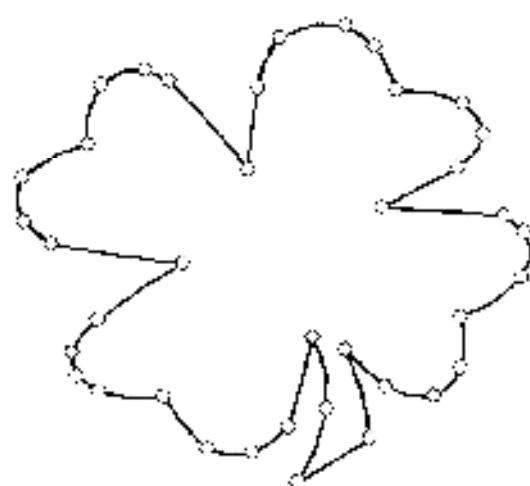
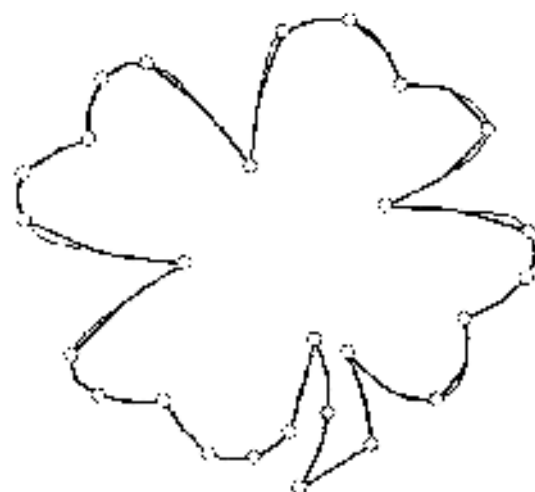
图 4.15 $T = 5$ 时的拟合情况及分界点图 4.16 $T = 12$ 时的拟合情况及分界点



图 4.17 原始形状 4

图 4.18 $T = 5$ 时的拟合情况及分界点图 4.19 $T = 12$ 时的拟合情况及分界点

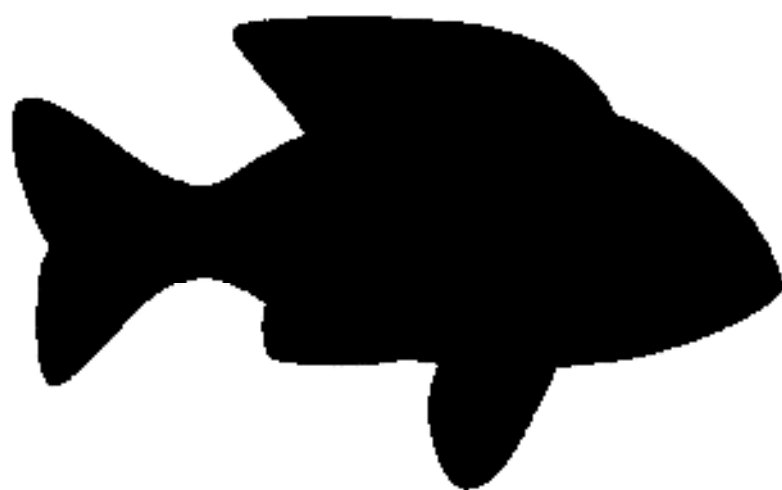
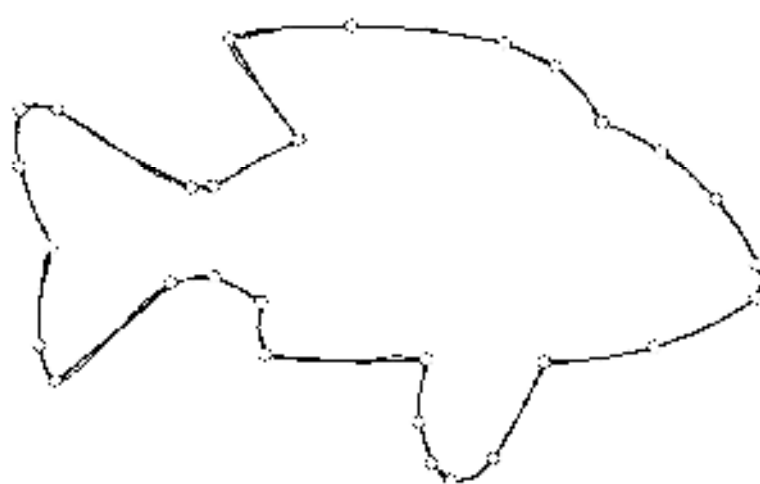


图 4.20 原始形状 5

图 4.21 $T = 5$ 时的拟合情况及分界点图 4.22 $T = 12$ 时的拟合情况及分界点

可以看出,当阈值 $T = 5$ 时,拟合曲线段已经能很好的和原始边缘重合,因此该误差能满足系统要求。

从压缩率来讲,当拟合精度满足要求时,即 $T = 5$ 时,以上各个图片中初始边缘点的个数与拟合后分界点与中点的数量及比值情况如表 4.2 所示:

表 4.2 各种形状的点的压缩情况

形状	原始边缘中点数	拟合后的分界点及中点数	压缩倍数
形状 1	1696	100	16.96
形状 2	1214	36	33.72
形状 3	868	18	48.22
形状 4	1373	76	18.07
形状 5	1473	58	25.40

因此,本算法在满足精度的同时能使数据量达到十几倍或几十倍的压缩,大大节约了存储空间。

4.4 本章小结

本章介绍了一种分段递归拟合的矢量化算法。该算法首先提取边缘中的尖点,然后对每两个尖点中的点序列用递归的方法进行拟合。总的来说,该算法在拟合精度和数据压缩率上基本都能满足鞋样系统的需求,且计算量相对现有算法要小。但目前该算法存在的缺点是递归拟合时对原始曲线都是利用中点进行简单分段,没有考虑到曲线自身的形状特征,因此最后得到的分界点还是有冗余,需要进一步改进。

第 5 章 结论

光栅图像的矢量化是一项综合性很强的技术,涉及到计算机图形学、图像处理、人工智能、CAD 技术等领域的知识,因此长期以来一直是一个技术上的难点。对于矢量化的研究,前人已经做了大量的工作,但仍然存在很多问题。本论文在研究前人工作的基础上,根据具体的项目要求,提出了一些自己的创新点,做出了一些改进,主要是以下两个方面:

1. 提出了一种间接的尖点提取方法:常规的尖点提取方法都是通过直接计算曲率,并根据其局部最大值求尖点。这些方法的缺点是算法复杂、计算量大。本文根据鞋样曲线闭合的特点,提出了先将曲线在不同方向上进行分解,得到两个一维的离散函数,然后用二阶差分代替曲率分别进行尖点提取,大大减少了计算量。

2. 提出了一种递归拟合的方法。在实际的应用中,矢量化算法并不是误差越小、压缩越多就越好,高精度和高压缩率往往需要大量的计算。本文提出的算法通过递归控制拟合的误差和压缩率,既能满足系统要求,又能减少不必要的计算,因此是一种较理想的算法。

以上两个算法实际应用于鞋样设计与扩缩 CAD 系统和农作物形态测量的项目中并取得了很好的效果。在总结项目经验的基础上,我根据以上两个创新点,在核心期刊上发表了两篇学术论文,分别是《一种间接提取轮廓特征点的方法》和《闭合数字曲线的多项式递归拟合》。

参考文献

- [1]苏丰. 图像图形处理与识别技术. 南京大学博士学位论文,2002
- [2]邹修明,张岳新.栅格地图矢量化关键技术研究与应用. 计算机工程与应用, 2003,19
- [3]T.P.Clement. The extraction of line-structured data from engineering drawings. *Pattern Recognition*,1981,14(1):43-52
- [4]C.Arcelli,and G.S.D.Baja. A thinning algorithm based on prominence detection. 1981, 13(3):225~235
- [5]Y.S.Chen,and W.H.Hsu. A modified fast parallel algorithm for thinning digital patterns. *Pattern Recognition Letters*.1988,7(2):99-106
- [6]A.Datta and S.K.Parui. A robust parallel thinning algorithm for binary images. *Pattern Recognition*,1994,27(9):1181-1192
- [7]D.T.Jasson, and A.M.Vossepoel. Adaptive vectorization of line drawing images. *Computer Vision and Image Understanding*, 1997,65(1):38-56
- [8]陈月林. 基于细化畸变校正的工程图智能识别方法的研究. 华中理工大学博士学位论文,1995
- [9]张树生,基于知识的细化畸变校正方法研究. 中国博士后首届学术大会论文集上册,1993
- [10]Monagan G, Roosli M. Appropriate Base Representation Using a Run Graph, *Proceeding 2nd International Conference on Document Analysis and Recognition*, Tsukuba, Japan,1993
- [11]陈勇,朱村,常明. 机械图纸中直线、圆弧的整体识别算法. 计算机辅助设计与图形学学报,1996,8 Supplement
- [12]李宾,谭建荣,彭群生. 基于逐步求精图像数据模型的矢量化方法研究. 计算机学报, 1996,19,Supplement.Special Issue of Chinagraph'96
- [13]徐建奎. 基于知识的机械图纸识别与理解方法的研究. 哈尔滨工业大学博士学位论文, 1994
- [14]D.Dori,W.Liu. Sparse pixel vectorization: an algorithm and its performance evaluation. *IEEETransaction on Pattern Analysis and Machine Intelligence*. 1999,21(3):202~215
- [15]张文景,许晓鸣,丁国骏,杨恩普. 一种基于曲率提取轮廓特征点的方法. 上海交通大学学报, 1999,33 (5)
- [16]刘楦,李佳,葛庆平.基于曲率计算的位图矢量化算法研究与应用.计算机工程与应用,

2002,38 (增刊)

- [17] 陈燕新, 戚飞虎. 一种新的提取轮廓特征点的方法. 红外与毫米波学报, 1998, 17 (3)
- [18] 张晓文, 杨煜普, 许晓鸣. 基于小波变换的特征构造与选择. 计算机工程与应用, 2003, 19: 25~28
- [19] 彭晓明, 张启衡. 一种基于小波变换的角点提取方法. 光电工程, 2000, 27(1)
- [20] Lee Jian-shu, et al. Multiscale corner detection by using wavelet transform[J]. IEEE Trans. Image Processing, 1995, 4(1): 100~104
- [21] 肖铁军, 丁明跃, 彭嘉雄. 基于迭代最近点的 B 样条曲线拟合方法研究. 中国图形图像学报, 2000, 5(7)
- [22] 李继云, 耿兆丰. 给定误差下最少数据点 B 样条曲线拟合的遗传算法研究. 计算机辅助设计与图形学学报, 2003, 15(3)
- [23] 涂嘉文, 徐守时, 谭勇. 基于最小均方误差的圆弧分段曲线拟合方法. 计算机应用, 2001, 21
- [24] 张习文, 李祐, 蔡士杰, 欧宗英. 基于遗传算法的以线段和圆弧为基元的曲线拟合. 计算机辅助设计与图形学学报, 2002, 14(2): 144~147
- [25] 郭子华, 庄镇泉. 基元遗传算法的地图矢量化算法. 中国科学技术大学学报, 1998, 28(4): 476~481
- [26] 郭子华, 蒙海华, 庄镇泉. 基于神经网络和遗传算法的地图自动输入新算法. 1998, 11(2): 190~197
- [27] Hongbin Zhang, Jiangjun Guo. Optimal Polygonal Approximation of Digital Planar Curves Using Meta Heuristics. Pattern Recognition, 2001, 34: 1429~1436
- [28] Yung-Nien Sun, Shu-Chien Huang. Genetic Algorithms for Error-bounded Polygonal Approximation. International Journal of Pattern Recognition and Artificial Intelligence, Vol. 14, No. 3: 297~314
- [29] 张纲要. 工程图纸矢量化技术研究. 西安建筑科技大学硕士学位论文, 2002
- [30] 章毓晋. 图像处理和分析, 图像工程上册. 清华大学出版社, 1999: 179~180
- [31] 冈萨雷斯. 数字图像处理 (第二版). 电子工业出版社, 2003: 485~486
- [32] 高等数学上册. 同济大学数学教研室编. 1996 年第四版: 392~396
- [33] C. H. Teh and R. T. Chin. On the detection of dominant points on digital curves. IEEE Trans. Pattern Anal. Mach. Intell. 1989: 11(8)

- [34] 肖轶军, 丁明跃, 彭嘉雄. 基于 B 样条模型的曲线特征点检测法. 数据采集与处理, 2000: 15(4)
- [35] 李庆扬, 王能超, 易大义. 数值分析. 清华大学出版社, 施普林格出版社. 2001
- [36] 孙兆林. MATLAB 6.x 图像处理. 清华大学出版社. 2002

攻读硕士学位期间参与的学术活动

一、发表的学术论文

(1) 刘玉兰, 葛庆平. 一种间接提取轮廓特征点的方法. 计算机工程与应用. 2004 年 4 月, 第 40 卷第 10 期.

(2) 刘玉兰, 葛庆平. 闭合数字曲线的多项式递归拟合. 计算机工程与应用. 2005 年 3 月, 第 41 卷第 7 期.

(3) 荆晶, 刘玉兰, 葛庆平. 曲线拟合中特征点的递归提取. 微机发展. (已录用, 2005 年 12 期或 2006 年 1 期发刊).

(4) 荆晶, 刘玉兰, 葛庆平. 正交多项式曲线拟合及在皮革自动切割中的应用. 计算机工程与应用(增刊). 2004 年第 40 卷.

二、参加的学术研讨会

(1) 2003 年 10 月 13 日~2003 年 10 月 15 日, 2003 年植物生长模型, 模拟可视化及其应用国际研讨会, 北京香山饭店.

(2) 2003 年 10 月 22 日~2003 年 10 月 24 日, 第二届智能化农业信息技术国际学术会议, 北京友谊宾馆.

(3) 2004 年 6 月 19 日~2004 年 6 月 21 日, 中国首届机器视觉技术及工业应用国际研讨会, 北京昌平小汤山.

三、参与的主要科研项目

(1) 鞋样设计和扩缩 CAD 系统, 温州市东方铆焊机械厂课题, 2004 年 7 月至 2005 年 7 月.

(2) 作物形态信息的计算机视觉检测技术研究, 市教委项目, 2003 年 1 月至 2005 年 12 月, 合同编号: KM200310028109.

(3) 作物图象测量标准与测量方法的研究, 国家 863 计划北京智能化农业信息技术研发中心课题, 2002 年 5 月至 2003 年 11 月, 课题编号 2001AA115190-08.

致 谢

感谢我的导师葛庆平老师。三年来，从进行课题研究到写学术论文、毕业论文，都是在葛老师的悉心指导下完成的。入学时，我还是一个不知道科学研究为何物的人，而如今我已具备了一定的科学研究能力，研究生期间也取得了一定的成果，而这一切，都是和葛老师的关怀教导分不开的。葛老师渊博的学识和一丝不苟的治学态度对我产生了很大的影响，从他那儿我不仅学到了知识，更学会了如何去当一名科学工作者。

感谢信息工程学院的各位老师给我的关怀和指导。

感谢我的同实验室的师哥师姐、师弟师妹们，在同他们的讨论和交流中使我见识大增，学到了很多书本上学不到的东西。在编程方面，徐歆凯和王文中同学给了我大量的帮助，非常感谢他们。

感谢我的父母、我的爱人轩波和我同宿舍的同学，他们在我遇到困难的时候给了我巨大的精神鼓舞和无微不至的关怀。

感谢所有帮助过和关心过我的老师、亲人和同学！