

西安邮电大学

毕业设计（论文）

题目： 基于拱序列的离散曲线匹配方法研究

学院： 自动化学院

专业： 自动化

班级： 自动 1403

学生姓名： 蔡宇杰

学号： 06141073

导师姓名： 邓颖娜 职称： 讲师

起止时间： 2017 年 12 月 5 日 至 2018 年 6 月 10 日

毕业设计（论文）声明书

本人所提交的毕业论文《基于拱序列的离散曲线匹配方法研究》是本人在指导教师指导下独立研究、写作的成果，论文中所引用他人的文献、数据、图件、资料均已明确标注；对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式注明并表示感谢。

本人完全理解《西安邮电大学本科毕业设计（论文）管理办法》的各项规定并自愿遵守。

本人深知本声明书的法律责任，违规后果由本人承担。

论文作者签名：

日期： 年 月 日

西安邮电大学本科毕业设计(论文)选题审批表

申报人	邓颖娜	职 称	讲师	学 院	自动化学院			
题目名称	基于拱序列的离散曲线匹配方法研究							
题目来源	科研				教学		其它	√
题目类型	硬件设计		软件设计	√	论文		艺术作品	
题目性质	应用研究		√		理论研究			
题目简述	<p>(为什么申报该课题)</p> <p>曲线匹配是计算机视觉和图像处理中的一个重要问题。判定几何图形形状相似性,得到与人类认知一致的结果,是目前的曲线描述与分类算法不能很好解决的问题。针对曲线匹配和几何图形形状的相似性判定,提出一种有效快速的基于拱序列的曲线匹配与相似性判定算法。</p>							
对学生知识与能力要求	<p>1.熟悉图像处理的基本知识;</p> <p>2.掌握 MATLAB 编程。</p>							
具体任务以及预期目标	<p>(应完成的具体工作,预期目标和成果形式)</p> <p>1.实现离散曲线的匹配方法设计;</p> <p>2.实现软件算法设计并编程实现,并进行仿真实验。</p>							
时间进度	<p>2017.12.5-2018.1.9: 查阅参考文献,撰写开题报告; 2018.1.10-2018.3.31: 设计算法思想流程; 2018.4.1-2018.4.30: 设计软件流程图并编写软件; 2018.5.1-5.30: 软件仿真与改进,撰写毕业论文; 2018.6.1-6.10: 打印论文,准备答辩。</p>							
系(教研室)主任签字	2017 年 12 月 9 日			主管院长签字		2017 年 12 月 9 日		

西安邮电大学本科毕业设计（论文）开题报告

学生姓名	蔡宇杰	学号	06141073	专业班级	自动 1403
指导教师	邓颖娜	题目	基于拱序列的离散曲线匹配方法研究		

选题目的（为什么选该课题）

物体的轮廓曲线是物体最直观的特征，信息量少但包含大量的有用信息。曲线匹配在文物碎片拼接、医学图像配准、产品检测等方面都有着广泛的应用。如何快速、准确地计算曲线相似性是计算机视觉和数字图像处理的一个重要问题。很多曲线匹配算法都能够在 MPEG7 等形状库上取得较好的分类结果，但是，在度量曲线相似性时，这些算法却往往不能获得与人类认知一致的结果。针对曲线匹配与相似性判定的准确性与快速性，打算采用一种基于拱序列的曲线匹配与相似性判定方法。利用角点信息，将曲线分割成子曲线，每两条子曲线组成一段拱，使用拱序列表示原曲线。每一段拱使用弓高弦长比、弧长弦长比等信息表示。利用拱的信息，可计算出两段拱的相似程度。利用字符串编辑距离的思想，计算出两个拱序列的相似度。基于拱序列的曲线匹配方法具有平移、旋转、缩放不变性，能够准确地计算出两条曲线的相似性。

前期基础（已学课程、掌握的工具，资料积累、软硬件条件等）

1、了解知识链码是一种常见的曲线描述方法。链码由 Freeman 等人提出，该方法将轮廓方向作为编码依据，用 4 个方向或者 8 个方向的 Freeman 链码表示轮廓曲线。后来出现了一系列改进的链码方法。链码能够简洁有效地表示形状，但是由于链码值的有限性，链码表示旋转的角度时存在一些问题。多边形近似是使用多边形来近似轮廓曲线，多边形近似的目的是使用尽可能少线段数来获取给定边界的基本形状。但是多边形近似很依赖于起始点的选择，且对于边界的扰动十分敏感。多边形近似并没有涉及曲线点与点之间的关系，Belongie 等人考虑了曲线点之系，提出了形状上下文的方法，对轮廓进行均匀采样，计算相对于每一个采样点，他点的分布，得到点的上下文特征表述。Li 等人提出一种多尺度形状上下文的方法，该方法使用尺度形状上下文描述符获得每个特征点的多个空间信息，该方法简单高效描述曲线，在等距 3 维形状的描述与检索中提升了性能。形状上下文融合了轮廓的局部信息和体信息，大幅提升了形状匹配的准确性。但是形状上下文不能很好地解决物体内部形变且计算的时间复杂度较高。史思琪等人根据局部轮廓曲率分布划分目标轮廓，并对步轮廓分段进行多级分段合并处理，得到一组完整描述多级目标特征的轮廓特征分段。同时，为了减少轮廓分段误匹配造成的分类错误，该算法完整描述目标特征从而提了目标识别准确率，但是对于细节较多的轮廓的表示，不够稳定。很多曲线匹配算法都能够在 MPEG7 等形状库上取得较好分类结果，但是，在度量曲线相似性时，这些算法却往往不能获得与人类认知一致的结果。准备条件已学《信号与系统》、《数字信号处理》、《模式识别技术应用》、《MATLAB》。熟练使用 MATLAB 2016a。查询 MATLAB 编程，曲线描述，拱序列，动态规划，形状相似性分析相关资料。可进行 MATLAB 编程的电脑以及软件 MATLAB 2016a。

要研究和解决的问题（做什么）

如何将曲线表示为拱序列？拱序列的定义，曲线转化为拱序列的途径。如何描述拱序列？拱描述子的概念与计算。如何实现对拱序列的匹配和相似性判定？拱序列的差异度判断，拱序列是否匹配的标准。如何验证这一算法是否正确？算法对图片的差异性判断，是否与人类视觉判定一致。什么是动态规划？动态规划的定义，动态规划方法的使用，以及使用时的注意事项。

工作思路和方案（怎么做）

具体方案如下：1.曲线预处理使用 Canny 算子对轮廓进行边缘提取，得到目标物体的轮廓曲线。曲线的小波动会导致提取错误角点，进而影响曲线的分割。为了避免提取错误角点，对于提取得到的曲线，进行 NURBS 曲线拟合，使用 de Boor 算法平滑曲线。2.提取特征点边缘方向发生急剧变化的地方称为角点，角点是图像的一个重要局部特征，具有旋转不变性。Harris 角点检测通过窗口在各个方向上的变化程度，决定是否角点。3.得到拱序列将曲线分割为一系列有序、相互重叠的短弧线，短弧线形状类似抛物线，命名为“拱”。4.拱描述子为了精确描述曲线，给出基于拱的曲线描述方法。对于一段拱，使用它的弓高弦长比、弓高半程弦长与弦长比、弧长弦长比、弓高点与弦中点连线与弦夹角的正弦值描述。5.两个拱之间的距离两个相似的拱具有相似的描述子，而不相似的拱的描述子差异度很大。基于拱描述子，可以得到两个拱差异度。给定拱 S_i 的描述子 $D_i = \{\alpha_i, \beta_i, \gamma_i, \theta_i\}$ 和拱 T_j 的描述子 $D_j = \{\alpha_j, \beta_j, \gamma_j, \theta_j\}$ ，定义差异度量函数 $D(S_i, T_j)$ ，差异度越小，拱越相似。显然，拱的差异度能很好地反映出拱是否相似。6.拱序列之间的相似度计算两条曲线的拱序列之间的差异度，作为两条曲线的距离值。拱序列之间差异度的计算方法，利用字符串编辑距离的思想，使用动态规划得到。编辑距离，是指两个字串之间，由一个转成另一个所需的最少编辑操作次数。编辑操作包括将一个字符修改成另一个字符，添加一个字符，删除一个字符。7.进行轮廓拼接实验、几何图形相似性的交叉度量实验和判定不同相似程度的形状对的实验来验证算法。

指导教师意见

签字：

2018 年 1 月 9 日

西安邮电大学毕业设计 (论文)成绩评定表

学生姓名	蔡宇杰	性别	男	学号	06141073	专业 班级	自动 1403
课题名称	基于拱序列的离散曲线匹配方法研究						
指导教师 意见	(从开题论证、论文内容、撰写规范性、学习态度、创新等方面进行考核) <div style="text-align: right;"> 评分(百分制): 指导教师(签字): _____ 年 月 日 </div>						
评阅 教师 意见	(从选题、开题论证、论文内容、撰写规范性、创新和预期成果等方面进行考核) <div style="text-align: right;"> 评分(百分制): 评阅教师(签字): _____ 年 月 日 </div>						
验收 小组 意见	(从毕业设计质量、准备、操作情况等方面进行考核) <div style="text-align: right;"> 评分(百分制): 验收教师(签字): _____ 年 月 日 </div>						
答辩 小组 意见	(从准备、陈述、回答、仪表等方面进行考核)						
评分比例	指导教师评分 (20%) 评阅教师评分 (30%) 验收小组评分 (30%) 答辩小组评分 (20%)						
学生总评 成绩	百分制成绩				等级制成绩		
答辩委员 会意见	毕业论文(设计)最终成绩(等级): <div style="text-align: right;"> 学院答辩委员会主任(签字、学院盖章): _____ 年 月 日 </div>						

目 录

第一章 引言.....	1
第二章 轮廓曲线特征提取.....	3
2.1 直接遍历法.....	3
2.2 八邻域边缘追踪法.....	3
第三章 角点特征提取.....	6
3.1 角点的定义.....	6
3.2 常见的角点检测算法.....	6
3.2.1 Harris 角点检测算法.....	6
3.4 基于轮廓曲线的角点检测算法.....	7
第四章 基于轮廓的拱序列特征提取.....	9
4.1 拱描述子.....	9
4.2 描述轮廓曲线.....	10
第五章 基于拱序列的曲线匹配.....	11
5.1 拱与拱之间的距离.....	11
5.2 字符串编辑距离.....	11
5.3 动态规划.....	11
5.4 拱序列之间的差异度.....	12
第六章 系统实现与实验结果分析.....	13
6.1 轮廓曲线提取.....	13
6.2 角点提取.....	14
6.3 曲线匹配.....	16
6.4 形状相似匹配.....	19
总结与展望.....	21
致 谢.....	22
参考文献.....	23
附录 1.....	24
附录 2.....	25
附录 3.....	27
附录 4.....	29

附录 5.....	32
-----------	----

摘 要

随着计算机视觉和图像处理的快速发展,曲线的描述和匹配逐渐成为一个重要的问题。物体的轮廓曲线是物体最直观的特征之一,通过对曲线的描述,得到曲线的特征,从而进行曲线之间的匹配。目前已经有很多种曲线描述和匹配的算法,但大多数都不可以很好的判定几何图形形状相似性,从而得到与人类认知一致的结果。

本文研究了一种基于拱序列的曲线匹配与相似性算法。首先提取物体的轮廓曲线,然后在曲线上提取角点,将曲线分割成一段段的拱,用拱序列来描述曲线。对于每一个拱,使用拱描述子来进行描述。最后通过拱描述子,来实现对曲线匹配与形状相似性的研究。经研究表明,这种算法可以实现对曲线的描述和匹配,判断几何图形的相似性。最终也可以得到与人类视觉判定基本一致的结果。

关键词: 轮廓提取; 曲线描述; 曲线匹配; 字符串编辑距离; 动态规划。

Abstract

With the rapid development of computer vision and image processing, the description and matching of curves has gradually become an important issue. The contour curve of an object is one of the most intuitive features of the object. Through the description of the curve, the characteristics of the curve are obtained, so that the matching between the curves is performed. There are already many kinds of algorithms for describing and matching curves, but there is no one that can well determine the similarity of geometric shapes and thus achieve results consistent with human cognition.

This paper introduces a curve matching and similarity algorithm based on arch sequences. Firstly, the contour curve of the object is extracted, then the corner point is extracted on the curve, the curve is divided into a section of arch, and the curve is described by the arch sequence. For each arch, use arch descriptors to describe. Finally, by using the arch descriptors, we can study the matching of curves and the similarity of shapes. The research shows that this algorithm can realize the description and matching of curves and judge the similarity of geometric figures. Eventually, results that are basically consistent with human visual judgments can also be obtained.

Keywords: contour extraction, curve description, curve matching, String programming ideas, dynamic programming.

第一章 引言

曲线，最直接的解释就是弯曲了的线，它定义在几何空间之中。每条曲线都有其曲率，曲率为 0 的线就是直线。曲线在不同几何空间中的定义有着极大的差别，本文的曲线定义在欧几里得几何空间中。

随着数字图像处理技术的发展，曲线匹配已经在很多方面得到了广泛的应用，例如医学图像的配准、文物碎片的拼接和产品的检测等各个领域。如何快速高效的对曲线进行匹配，已经成为我们面临的一个重要而且必须解决的问题。

曲率是曲线最主要的特征之一，基本可以代表一条曲线，不同曲线的曲率不同。根据曲率我们可以比较一些较为规则的曲线，得到它们之间差异程度的大小。但当曲线较为复杂的时候，我们就需要使用其他方法。

曲线的匹配是基于对曲线的描述的，所以对曲线的描述不同的话，曲线的匹配也会有所不同。现在已经有了很多种对曲线进行描述的方法。

Freeman 等人^[1]提出的链码就是一种常见的曲线描述方法。这种方法的编码基础是轮廓方向，轮廓曲线的表示用 4 个方向或者 8 个方向的 Freeman 链码来进行。后来对于这种方法，很多人进行了改进，得到了效果更好的方法。链码可以简单有效的表示物体的形状，但是在表示旋转角度时一直存在一些问题。

也有很多人基于多边形近似提出了几种曲线描述的算法，每种都可以在某方面得到非常好的效果。基于局部积分偏差的多边形近似算法，达到了良好的多边形近似效果，它是 Ramaiah^[2]等人创造并验证的。而对多边形的分割点进行修正^[3]，使得在基元分割中取得良好的效果，李欣言等人完成了这些工作。

史思琪等人^[4]基于局部轮廓的曲率分布来划分目标的轮廓，并对初步的轮廓分段进行多级分段合并处理，从而得到一组完整地描述多级目标特征的轮廓特征分段。与此同时，为了减少轮廓分段不正确匹配引起的在分类上的错误，该算法通过完整地描述目标特征，从而提高了对目标识别的准确率，但是在细节较多的轮廓的表示方面，效果还不够稳定。

本文通过拱描述子所描述的一段段小拱，组合起来形成一条曲线。所以也可以通过一些方法将曲线分解为小拱，这些小拱的集合就是对曲线的描述。分解的拱越多，对曲线的描述就会越准确。

描述曲线之后下一步就是对曲线进行匹配。近年来，越来越多的曲线描述和匹配方法被国内外众多专家所提出。这些方法在很多方面都可以取得较好的效果。经过多年的研究，曲线的匹配方法现在已经有了非常多的种类。

Sebastian^[5]等人利用曲线的曲率和弧长进行曲线的匹配。Liu 等人创造性的用形状特征来构建相似度矩阵，最后求解矩阵的最优匹配结果。Cohen 等人首先通过对顶点的切矢做内积处理，然后构造度量函数，最终实现对曲线的匹配。

但大多数算法在实现几何图形形状匹配时，与人眼视觉得到的结果有所差

异。所以为了解决这一问题，本文介绍一种新的曲线描述和匹配算法，这种算法是基于拱序列的。先利用八邻域边缘追踪算法得到轮廓曲线，再使用 Harris 算法提取轮廓曲线的角点。由于 Harris 算法提取的角点不太理想，本文使用了基于轮廓曲线的角点检测算法来提取角点。各个角点把轮廓曲线分为一段段拱，通过一系列计算，用拱描述子来描述每一个拱。最后根据拱描述子来构建相似性度量函数，使用动态规划的方法来计算差异度，从而实现曲线的匹配。

基于拱序列的曲线匹配方法拥有平移、旋转、缩放不变性，能够快速、准确地计算出曲线的相似性，从而实现曲线的匹配。并且在几何图形形状相似性方面也非常敏感，几何图形轮廓的不同会导致角点的位置和数量都可能发生变化，根据本文算法，会导致匹配度的值上升。

第二章 轮廓曲线特征提取

轮廓曲线是指给定图片，对图片进行二值化处理后，图像的内外轮廓的曲线。本文中的轮廓曲线只指外轮廓曲线，也可以说是对图像的边缘进行追踪所提取出来的曲线。关于轮廓曲线提取的方法已经有了很多，例如基于细胞神经网络的边缘提取算法^[6]、基于方向形态学和数学形态学的轮廓提取法等，常用的轮廓曲线提取方法包括直接遍历和八邻域边缘追踪^[75]等。

2.1 直接遍历法

对于轮廓曲线的提取，最为简单的方法就是直接遍历法。先将图像转换为二值图，从上下左右四个方向分别遍历各个点，遇到值为 1 的点停止遍历。通过对四个图像矩阵的求和来合并图像，最终结果就是轮廓曲线的图像。这种方法简单且便于理解，对于大多数较为规则的图像都可以准确提取出其轮廓曲线。但在一些特殊情况下，会丢失部分轮廓，导致结果不准确。例如如图 2-1 所示在人物头像中上方头发处内凹的曲线无法遍历到，就会丢失这段轮廓，最终导致结果错误。



图 2-1 头像图

2.2 八邻域边缘追踪法

以一个像素点为中心，建立一个 3*3 的方格，如图 2-2 所示，在像素点周围有八个邻域点，这八个点均与中心像素点相邻，以中心点为原点，建立笛卡尔坐标系，可得由 1 至 8 邻域点的坐标分别为 $(-1, 0)$ ， $(-1, 1)$ ， $(0, 1)$ ， $(1, 1)$ ， $(1, 0)$ ， $(1, -1)$ ， $(0, -1)$ ， $(-1, -1)$ 。

2	3	4
1	中心点	5
8	7	6

图 2-2 八领域图

八领域边缘追踪法首先随机选取轮廓曲线上的一个点作为初始点，然后以该点为中心点，找出其八领域各点，设置初始方向，从初始方向开始顺时针遍历八领域各点，遇到值为 1 的点时停止遍历。以这个值为 1 的点再次建立八领域，初始方向为前一个点到这个点的方向逆时针旋转 90 度，再次顺时针遍历八领域各点，找到下一个值为 1 的点。以此循环下去，直到找到的下一个点为初始点时停止循环。将得到的点全部存入另一个全 0 图像矩阵，则该图像矩阵的图像就是提取出来的轮廓曲线图像。

由于八领域边缘追踪法是根据上一个点来找下一个点，所以可以准确提取所有轮廓曲线，在轮廓曲线上内凹的曲线也可以找到，这种方法适用于绝大多数图像的轮廓提取。但是当轮廓出现断点时，如图 2-3 所示，这种方法就会陷入死循环，无法提取轮廓曲线。



图 2-3 有断点的轮廓图

插值，又称内插，是根据已知离散数据点，来计算新数据点的一种方法。最

简单的插值方法是最近邻插值，它只需要找到最近的数据值，并分配一样的值。此外，线性插值也是一种非常简单的方法，它通过计算断点周围两点所连直线的斜率，断点和其中一点所连直线的斜率与之相同，来确定断点位置，赋予两点的平均值。但是当要在高纬度的多变量插值时，最近邻插值要比线性插值速度更快更简单。

对于以上问题，可以通过使用合适的方法给断点处插值来解决，这样就可以正常的提取出轮廓曲线。曲线提取出来后，可以去掉断点处插入的值，最终得到正确的轮廓曲线。但本文所描述的算法要求轮廓曲线必须连续，不能出现断点，否则无法继续进行之后的一系列运算，不能对轮廓曲线进行匹配。

由于八领域边缘追踪法的准确性，在本文中提取轮廓曲线时使用了八领域边缘追踪法，通过对图像边缘的精准追踪，来得到完美的外轮廓曲线图。使得后面的计算误差减小，匹配结果更加优秀。

第三章 角点特征提取

3.1 角点的定义

一般的角点检测都是检测有具体定义的、或者是能够具体检测出来的兴趣点的。所以兴趣点或许是角点，也可能是完全孤立的点、线段两端的端点，又或者是曲线上局部曲率最大的点。在真正检测的时候，众多角点检测的方法检测的往往都是兴趣点，而不是真正的角点。所以，当只需要保留真正角点的时候，不可避免要对检测出来的伪角点即兴趣点进行处理，去除所有伪角点，剩下的就是可以使用有效角点。角点数量较少的情况下，或许可以得到所有有效角点。但是当角点数量较为庞大的时候，很难有一种方法可以完美去除所有伪角点。所以针对具体情况，需要具体分析，选择最合适的方法来消除伪角点。

本文对角点的定义为：在轮廓边缘上方向发生急剧变化的点就是角点。角点对图像图形的分析和处理有着相当重要的意义，是图像的重要特征之一。角点常常应用于图像的匹配，可以有效的描述图像的一部分特征。

3.2 常见的角点检测算法

基于灰度图像的角点检测方法、基于二值图像的角点检测方法和基于轮廓曲线的角点检测方法是三类目前最主要的角点检测算法。基于灰度图像的角点检测方法根据又梯度、模板和梯度模板组合三种方式形成了三种方法，其中基于模板的方法主要考虑图像的亮度变化，即像素领域点的灰度变化，其角点定义为与相邻点灰度值发生剧烈变化的点。Kitchen-Rosenfeld 角点检测算法、KLT 角点检测算法、Harris 角点检测算法^[8]及 SUSAN 角点检测算法是四种常见的基于模板的角点检测算法。相对于另外的角点检测算法，SUSAN 角点检测算法的算法更加简单、位置更加准确、抗噪声能力更加强大。

3.2.1 Harris 角点检测算法

它是一种典型的基于灰度图像的角点检测。Harris 角点检测算法的基本原理如下：

在图像上设置一个局部的小窗口，朝各个方向移动这个小窗口，如果窗口内的灰度值发生剧烈变化，则在这个窗口内有角点存在。当小窗口朝各个方向移动时，灰度值均无变化，就认为小窗口中不存在角点；若小窗口朝某个方向移动时，灰度值有较大的变化，而朝其他方向移动时灰度值无变化，就认为小窗口中可能有一条直线的线段。如图 3-1 判断图所示。

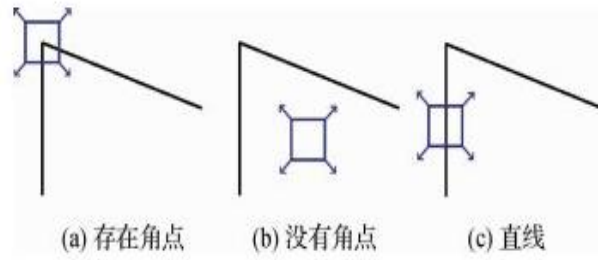


图 3-1 判断图

Harris 角点检测主要由以下 4 步组成：

（1）利用水平差分算子 $f_x = [-2 \ -1 \ 0 \ 1 \ 2]$ 和竖直差分算子 $f_y = [-2; -1; 0; 1; 2]$ 对图像上每一个像素点进行滤波处理，从而求得 I_x^2, I_y^2 和 $I_x * I_y$ 的值，来构建矩阵

$$M = \begin{bmatrix} I_x^2 & I_x * I_y \\ I_x * I_y & I_y^2 \end{bmatrix};$$

（2）建立高斯窗函数，对矩阵 M 中的 I_x^2, I_y^2 和 $I_x * I_y$ 进行高斯平滑滤波，得到新的 I_x^2, I_y^2 和 $I_x * I_y$ 的值，构建新的矩阵 M ；

（3）根据新的矩阵 M ，求出每个像素的角点响应函数 R 的值。响应函数

$R = \det(M) - 0.08 * (\text{trace}(M)^2)$ ，求出最大的 R 值即 R_{\max} ，然后进行局部极大值抑制。

（4）遍历所有像素点，如果像素点的 R 值大于其八邻域各点的 R 值且大于阈值 $0.01 * R_{\max}$ ，则判定该点为一个角点。

Harris 角点检测算法的优点是计算简单，便于理解，高度自动化，适用于大多数灰度图的角点检测。但是这种算法不具备尺度不变性，而且提取的角点是像素级的，容易提取出很多错误角点，对图片的质量有较高的要求。

3.4 基于轮廓曲线的角点检测算法

这种算法^[9]也是基于像素点，首先在轮廓曲线上找到一个初始点，求初始点相邻两点，两点分别连接初始点，如果这三点不在同一条直线上，则初始点为角点，否则不是。然后在以初始点右边的点为初始点，判断它是否为角点，依次循环，直到下一个点是最初的初始点时停止。这时会找出非常多的角点，其中相对于人眼视觉有一些是错误的。所以，再从找到的角点中找一个初始角点，分别连接它左右两个相邻角点，然后求得所形成的夹角的大小，求夹角时要连接相邻角点，形成三角形，用勾股定理计算出夹角。此时要设置一个合适的角度，当该角度大于夹角的时候，初始角点为真角点，否则是伪角点。以此方法分别判断所有角点是否为真角点，从而得到最终的角点集合。

如果设置的角度合适的话，这种算法可以排除大多数假角点，得到一个较为准确的结果，而且计算较为简单，便于理解。但是要得到合适的角度，需经过大量实验，进行对比，耗时耗力。根据伪角点的数量和位置的不同，可以连接角点左右相隔一个或数个角点的角点，来计算夹角，排除伪角点，从而得到更加准确的有效角点。

对于一些简单的，有效角点和伪角点数量都比较少的轮廓曲线，可以根据观察直接排除伪角点，这样更加简单，而且准确。当轮廓曲线的有效角点的数量较少，而伪角点的数量较多时，可以先后使用不同的算法，不同的角点间隔数量来去除伪角点，得到准确的有效角点。

第四章 基于轮廓的拱序列特征提取

4.1 拱描述子

角点提取完成后，设置一个初始角点。从初始角点开始，每两个点将轮廓曲线分割为一段曲线，最后一个角点与初始角点之间为最后一条曲线。 n 个角点将轮廓曲线分为 n 段曲线，相邻的两条曲线组合为一个拱，共有 n 个拱。所以可以用含有 n 个拱的拱序列来表示拥有 n 个角点的轮廓曲线。

观察图 4-1 拱图中的拱可以发现，这个拱本身就不是规则的，无法用一个曲率来描述，甚至每个像素点的曲率都各不相同。所以对于这类拱所组合的曲线，我们无法用曲率进行描述，必须使用其他的方法。

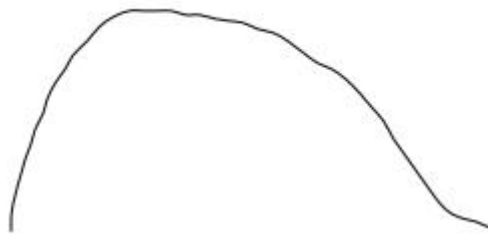


图 4-1 拱图

每个拱都可以用一组拱描述子^[10]来描述，其中包括弓高弦长比 a 、弓高半程弦长与弦长比 b 、弧长弦长比 c 、弓高点与弦中点连线与弦夹角正弦值 d 。如图 4-2 示例图所示，计算方法如下：

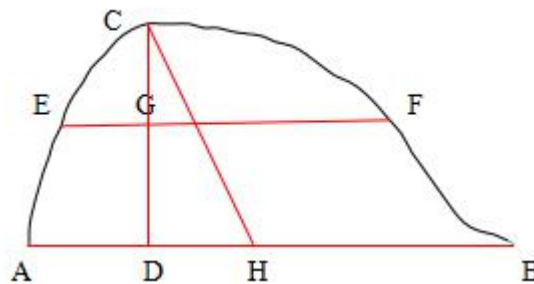


图 4-2 示例图

- (1) 连接 AB, 找拱上离 AB 最远一点为 C, 过 C 作 CD 垂直于 AB, 交点为 D, 则弓高弦长比 a 为 CD 与 AB 长度之比;
- (2) 取 CD 中点为 G, 过 G 点作 AB 的平行线, 与拱交点分别为 E、F, 则弓高半程弦长与弦长比 b 为 EF 与 AB 的长度之比;
- (3) 计算拱的长度 s , 则弧长弦长比 c 为 s 与 AB 的长度之比;
- (4) 取 AB 中点为 H, 连接 CH, 则弓高点与弦中点连线与弦夹角正弦值 d 为

$\angle AHC$ 的正弦值，是 CD 与 CH 的长度之比。

综上所述，这个拱的拱描述子 D 为 $\{a,b,c,d\}$ ，可以准确描述这个拱。

拱描述子是基于拱本身求得的，所以它不会随着拱的平移、旋转、缩放而发生改变，在这个方面远远优于其他的描述方法。此外，它所需要的计算并不算特别多，耗时也不算特别久，而且内容并不复杂，理解起来也很容易。

4.2 描述轮廓曲线

经过以上工作，我们知道可以用含有 n 个拱的拱序列来表示拥有 n 个角点的轮廓曲线，而每个拱都可以用一组拱描述子来描述。设 $S=\{S_i|i=1,2,\dots, m\}$ 和 $T=\{T_j|j=1,2,\dots, n\}$ 为两条待匹配的轮廓曲线的拱序列。拱序列 S 的第 i 个拱用 S_i 来表示，拱序列 T 的第 j 个拱用 T_j 表示。

第五章 基于拱序列的曲线匹配

5.1 拱与拱之间的距离

拱是由拱描述子来描述的，所以拱描述子相似，则两个拱相似，拱描述子差异过大，两个拱差异也大。综上所述，我们可以基于拱描述子来构建一个函数，来描述拱之间的差异程度。给定拱 S_i 的拱描述子 $D_i=\{a_i,b_i,c_i,d_i\}$ ，给定拱 T_j 的拱描述子 $D_j=\{a_j,b_j,c_j,d_j\}$ 。构建的差异度度量函数为：

$$D(S_i, T_j) = \frac{|a_i - a_j|}{\min(a_i, a_j)} + \frac{|b_i - b_j|}{\min(b_i, b_j)} + \frac{|c_i - c_j|}{\min(c_i, c_j)} + \left| \frac{d_i}{d_i + 1} - \frac{d_j}{d_j + 1} \right| \quad (5-1)$$

拱描述子不受拱平移旋转的影响，所以差异度度量函数的值也不受拱平移旋转的影响，得到的值较为可靠，可以良好的描述两个拱之间的差异度。

5.2 字符串编辑距离

已知如何计算两个拱之间的差异度，可以分别计算两个拱序列中所有拱之间的差异度，最终得到两个拱序列之间的差异度。这里我们要用字符串编辑距离的思想。

当我们想把一个字符串变为另一个字符串的时候，我们可以使用的编辑操作分别为：

- (1) 将一个字符修改为另一个字符；
- (2) 在字符串中插入一个字符；
- (3) 在字符串中删去一个字符。

两个字符串之间的编辑距离，即把一个字符串转变为另一个字符串所需的最小编辑操作次数^[1]。在一般的情况下，如果编辑距离越小，那么这两个字符串越相似，这就是字符串编辑距离的思想。这种思想常常被用于构建相似度计算函数，从而实现实际应用。

5.3 动态规划

根据之前字符串编辑距离的思想，通过使用动态规划的方法来计算拱序列之间的差异度。

动态规划作为现代运筹学的一个重要分支，是在求解决策过程(decision process)时优化度最好的数学方法。在公元一九五零年初的时候，美国数学家 R.E.Bellman 等人在研究多阶段决策过程(multistep decision process)的优化问题时，提出了著名的最优化原理(principle of optimality)，把多阶段过程转化为一系列单阶段问题，利用各阶段之间的关系，逐个求解，创立了解决这类过程优化问题的新方法——动态规划。他的著作《Dynamic Programming》，该领域的第一本著作，于公元一九五七年出版。

动态编程算法通常用于解决某些最佳性能的问题^[12]。在这类问题中，可能有很多可行的解决方案。每个解决方案都对应一个值，我们希望找到具有最佳价值的解决方案即最优的值。动态规划算法类似于分治法。它也是将待求解问题分而治之，先分为数个子问题，先解决子问题，在将子问题的解决结果结合起来，得到总问题的解决结果。但当经过分解所得到的子问题不相互独立的时候，分治法将不再适用，而动态规划可以解决这一问题。动态规划法的基本思路是先记录下所有已经解决的子问题的答案，再根据需要找到已经记录下来的答案。如果使用分治法的话，在分解之后会产生大量重复出现的子问题，计算量剧增，浪费时间。所以我们可以将所有已解的子问题的答案保留在一张表之中。无论单个子问题是否在后面的计算过程中是否调用，只要计算出它的答案，就将其填入表中。动态规划的算法有很多，而且各不相同，但它们的填表格式是一样的。

5.4 拱序列之间的差异度

已知如何计算两个拱之间的差异度，可以分别计算两个拱序列中所有拱之间的差异度，最终得到两个拱序列之间的差异度。这里我们可以用字符串编辑距离的思想,通过使用动态规划的方法来计算拱序列之间的差异度。

将拱 S_i 和拱 T_j 之间的差异度的值作为将拱 S_i 修改为拱 T_j 所需要的代价，设添加拱和删除拱所需要的代价为 c 。利用动态规划的方法，将拱序列之间的差异度表示在一个 $m+1$ 行， $n+1$ 列的矩阵中。

设 $n(i,j)$ 的值为拱序列 $S=\{S_i|i=1,2,\dots, m\}$ 前 i 个拱和拱序列 $T=\{T_j|j=1,2,\dots, n\}$ 前 j 个拱的最小编辑距离， $n(i,j)$ 的值赋予矩阵第 i 行第 j 列的数。 $n(i,j)$ 的计算方法如下：

当 $i=0,j=0$ 时， $n(i,j)=0$;

当 $i=0,j>0$ 时， $n(i,j)=n(i,j-1)+c$;

当 $i>0,j=0$ 时， $n(i,j)=n(i-1,j)+c$;

当 $i>0,j>0$ 时，如果将拱 S_i 修改为拱 T_j ，则 $n(i,j)=n(i-1,j-1)+D(S_i,T_j)$ ；如果增加拱 S_i ，则 $n(i,j)=n(i-1,j)+c$ ；如果增加拱 T_j ，则 $n(i,j)=n(i,j-1)+c$ ；由于 $n(i,j)$ 的值为最小编辑距离，所以综上所述， $n(i,j)=\min\{n(i-1,j-1)+D(S_i,T_j), n(i,j-1)+c, n(i-1,j)+c\}$ 。

$S=\{S_i|i=1,2,\dots, m\}$ 和 $T=\{T_j|j=1,2,\dots, n\}$ 这两条待匹配的轮廓曲线的拱序列得最小编辑距离为 $n(m,n)$ ，定义这两个拱序列之间的距离为 $G(S,T)=n(m,n)/(m+n)$ 。则匹配度为 $G(S,T)$ 。如果 $G(S,T)$ 越小，则说明两条轮廓曲线之间的相似度越高。所以设置一个适当的阈值 G ，若 $G(S,T)<G$ ，则两条轮廓曲线匹配，否则不匹配。

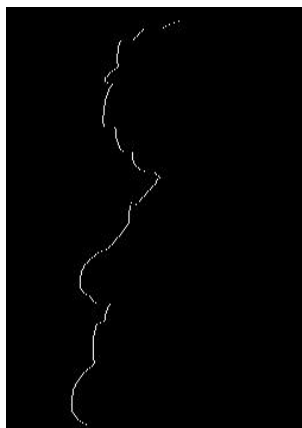
第六章 系统实现与实验结果分析

6.1 轮廓曲线提取

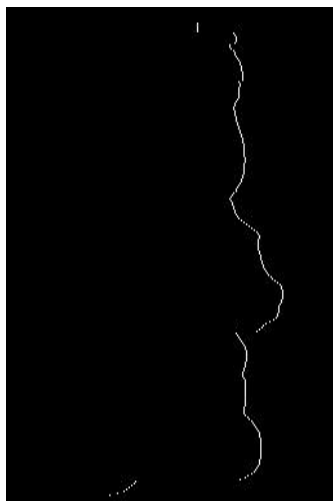
我对图像轮廓曲线的提取做了实验，以图 6-1 原图为例，使用直接遍历法和八邻域边缘追踪法分别提取轮廓曲线。图 6-2（a）（b）（c）（d）分别为从四个方向上遍历所得的轮廓图，图 6-3 为直接遍历法最终提取的轮廓曲线图。具体代码见附录 1。图 6-4 为八邻域边缘追踪算法所求得的轮廓曲线图。具体代码见附录 2。



图 6-1 原图



(a)



(b)



(c)



(d)

图 6-2 四方向轮廓图 (a) (b) (c) (d)

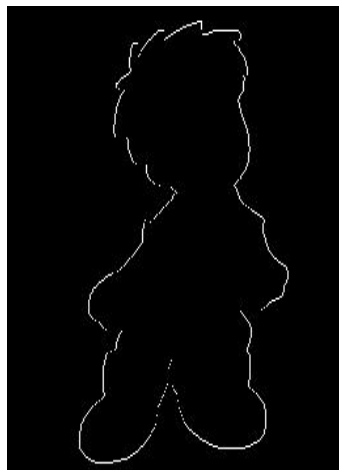


图 6-3 轮廓图

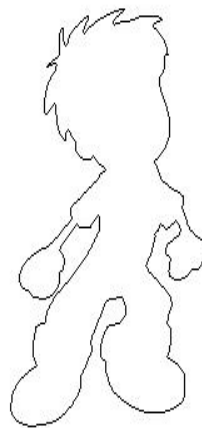


图 6-4 轮廓曲线图

图 6-1 原图边缘非常复杂，轮廓较难提取。对比图 6-3 和图 6-4 可以发现，直接遍历法虽然大致轮廓已经提取完成，但是在提取过程中还是丢失了部分轮廓曲线。这是因为这种算法在遍历到真正轮廓点外侧的点时就会停止，导致轮廓点丢失。而八邻域边缘追踪法则是根据一个轮廓点提取下一个相邻的轮廓点，保证不会丢失点，最终完美的提取了全部的轮廓曲线，效果非常好。本文在提取轮廓曲线中采用了八邻域边缘追踪法。

6.2 角点提取

得到轮廓曲线后，下一步就是提取特征点，也就是角点。我对角点提取也做

了实验，以图 6-4 为例子，分别使用 Harris 角点检测算法和基于轮廓曲线的角点检测算法提取角点。Harris 角点检测算法提取的角点图如图 6-5 所示。由于本文只需要轮廓曲线上的角点，所以对角点进行处理，最终结果如图 6-6 所示。具体代码见附录 3。基于轮廓曲线的角点检测算法提取的角点图如图 6-7 所示。由于这种方法是基于轮廓曲线的，所以角点全在轮廓曲线上。具体代码见附录 4。

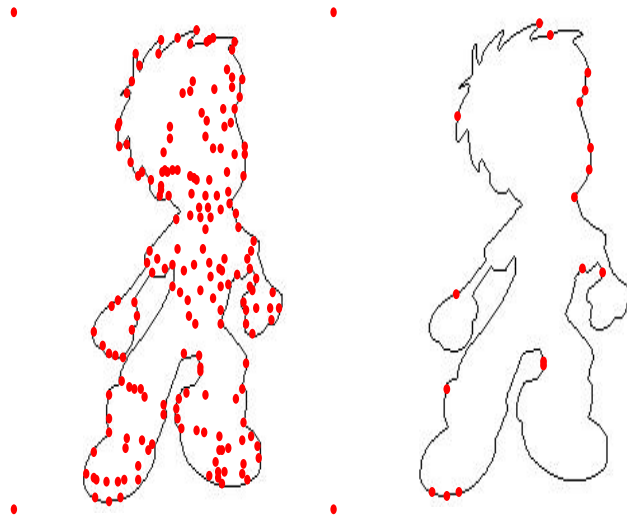


图 6-5 角点图（Harris 角点检测算法）

图 6-6 处理后的角点图

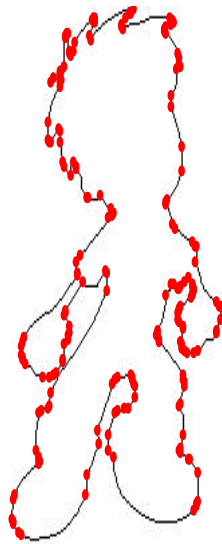


图 6-7 角点图（基于轮廓曲线的角点检测算法）

由图 6-5 和图 6-6 可知，Harris 角点检测算法提取的角点数量很多，而且大多数都没有落在轮廓曲线上，无效角点过多，有效角点数量不足。这是因为响应函数 R 中有一个不定值 k ，本文取 $k=0.08$ 。但是 k 值直接影响角点的提取和有效角点的数量，需要进行大量实验选取最佳 k 值。

由图 6-7 可知，基于轮廓曲线的角点检测算法提取的角点大都为有效角点，只有部分角点距离太近，可以选择性舍去数个，保留一个即可。因为这种算法是基于轮廓曲线的相邻轮廓点的位置变化来判断角点的，如果轮廓边缘的点分布不够平滑，就会导致错误角点的出现。本文在角点检测时使用了基于轮廓曲线的角点检测算法。

6.3 曲线匹配

最后做基于拱序列的曲线匹配实验，首先将图 6-9 与自己进行曲线匹配，计算匹配度。在把图 6-8 轮廓图和图 6-10 轮廓图分别与图 6-9 轮廓图进行曲线匹配，计算匹配度。实验结果如表 6-11 所示，具体代码见附录 5。



图 6-8 轮廓图

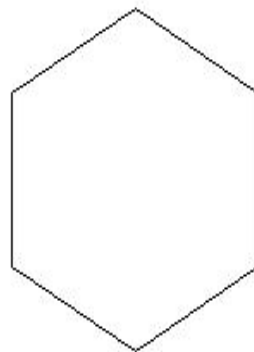


图 6-9 轮廓图

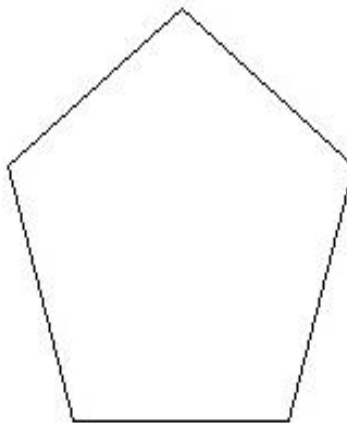


图 6-10 轮廓图

表 6-11 匹配度结果表

	图 6-9	图 6-8	图 6-10
图 6-9	0	5.3134	1.3132

由表 6-11 可知，匹配度的值越小，则轮廓曲线的相似性越高。实验结果表面，相对于图 6-8，图 6-10 与图 6-9 更相似。实验结果与人眼视觉所得结果相同。本算法流程框图如图 6-12 所示。

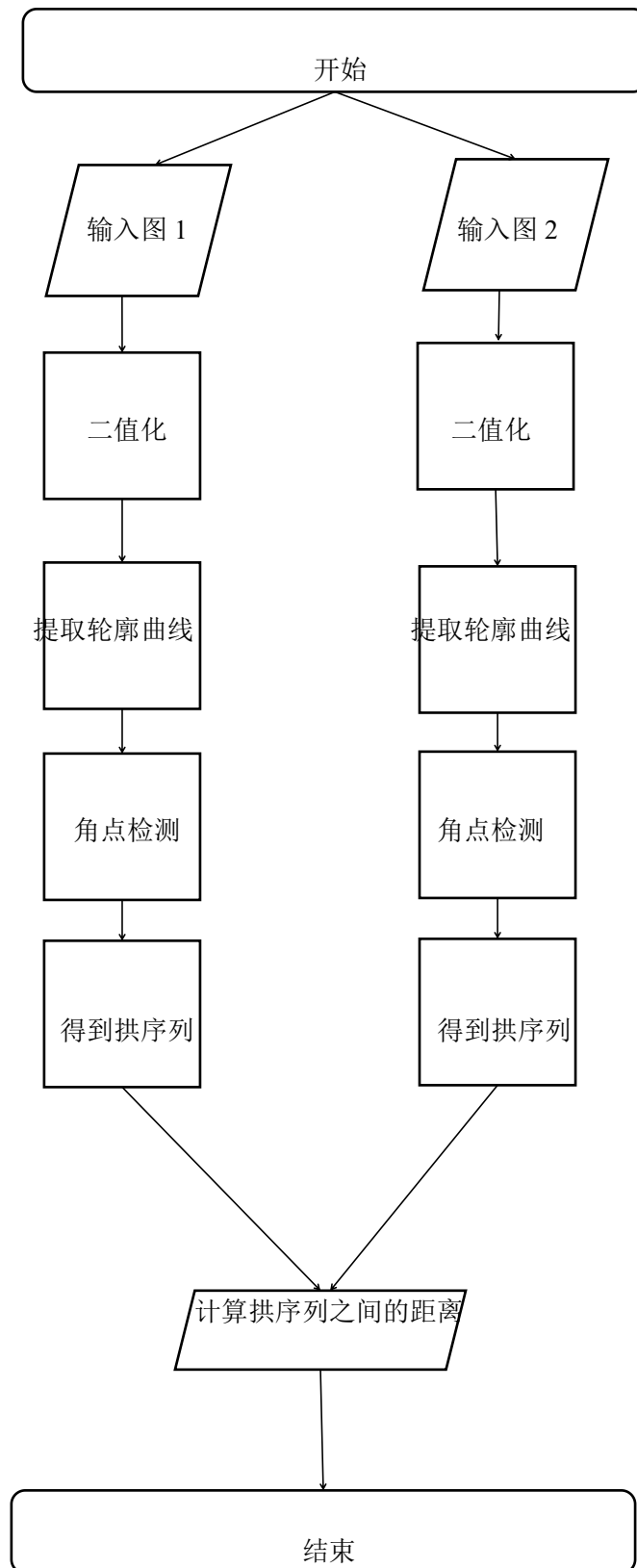


图 6-12 流程框图

6.4 形状相似匹配

本文算法相对于其他曲线匹配的算法，主要优势在于对轮廓形状的敏感性，所以设计一个实验来验证。取三对形状相似的图像，如图 6-13（a）（b），图 6-14（a）（b），图 6-15（a）（b）所示，分别计算它们之间的匹配度，结果如表 6-16 所示。

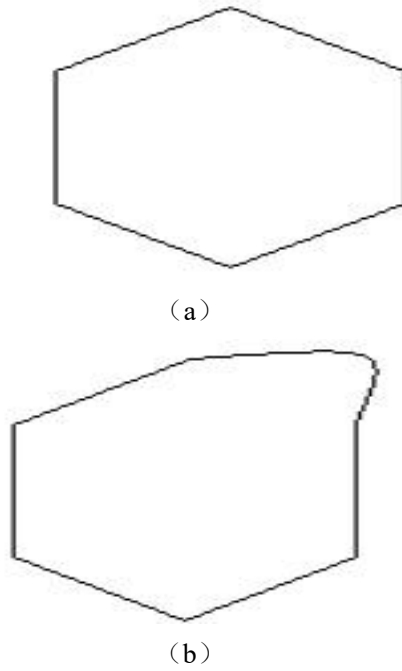
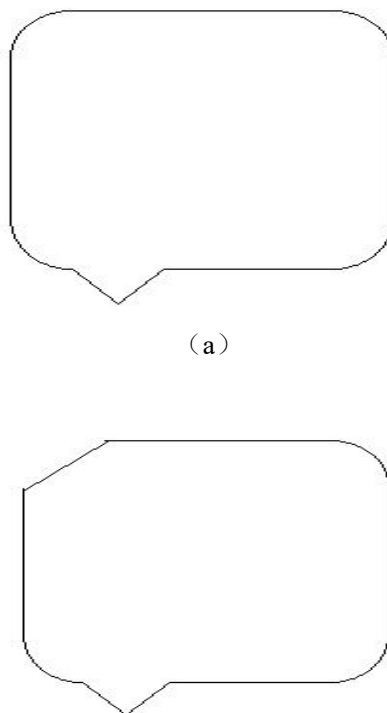
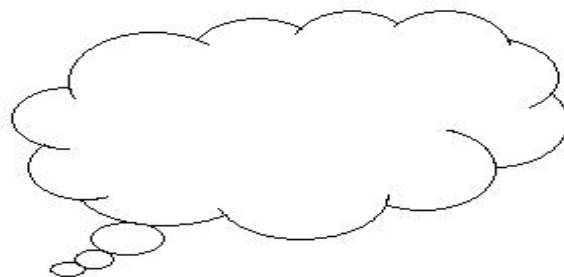


图 6-13

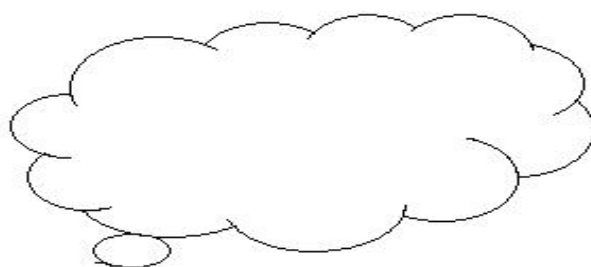


(b)

图 6-14



(a)



(b)

图 6-15

表 6-16 匹配度结果表

	图 6-13 (a) (b)	图 6-14 (a) (b)	图 6-15 (a) (b)
匹配度	1.5327	1.0646	0.7771

表 6-16 证明，本文算法对轮廓曲线形状的较小差异非常敏感。两条曲线的长度和角点数量对本文算法所求匹配度有很大的影响，如果轮廓曲线上的一段发生变化，那么曲线的长度和角点也会变化，直接导致匹配度的上升，所以基于拱序列的曲线匹配对轮廓曲线形状的较小差异较为敏感。

总结与展望

曲线的描述与匹配是计算机视觉与数字图像处理中的一个必不可少的课题，本文所记述的基于拱序列的曲线匹配与相似性算法，可以有效完成曲线的描述与匹配，并且基本可以得到与人眼视觉判断所一致的结果。但是现在还没有一种效果非常好的角点检测算法，可以对基本所有图像的轮廓曲线提取出全部有效的角点。所以今后可以继续研究如何实现可以达到这种效果的角点检测算法。

此外，本文算法可以计算出匹配度的值，今后可以研究如何设置一个合适的标准匹配度的值，大于该值则匹配，否则不匹配。

致 谢

转眼四年已经过去，大一时稚嫩的我们初次踏入大门的场景仿佛就在昨日。经过数月的努力，我在指导老师的帮助时成功完成了毕业设计。在此期间，我不仅复习了之前学过的知识，还知道了很多新的东西。

在此首先感谢我的毕业设计指导老师，西安邮电大学自动化学院的邓颖娜老师，我的每一点进步都离不开她的教导，在这里我要说一句“老师辛苦了”。此外还要感谢与我讨论的同学们，在讨论中选择最适合的方法，完成自己的设计。最后谢谢各位答辩老师，金无足赤，本论文可能会有所不足，希望能得到各位老师的指导和建议。

参考文献

- [1] Freeman H. Techniques for the digital computer analysis of chain-encoded arbitrary plane curves[C]//Proceedings of National Electronics Conference. Berlin: Springer International Publishing, 1961: 421-432. [DOI:10.1007/BFb0041115]
- [2] Bribiesca E, Guzman A. How to describe pure form and how to measure differences in shapes using shape numbers[J]. Pattern Recognition, 1980, 12(2): 101-112. [DOI:10.1016/0031-3203(80)90009-6]
- [3] 李欣言, 杜文华, 王俊元, 等. 一种平面轮廓分割与基元识别方法研究[J]. 制造技术与机床, 2016(1): 85-88. [DOI:10.3969/j.issn.1005-2402.2016.01]
- [4] 史思琦, 石光明, 齐飞. 基于特征完整描述的部分遮挡目标识别算法[J]. 系统工程与电子技术, 2011, 33(4): 913-918. [DOI:10.3969/j.issn.1001-506X.2011.04.40]
- [5] Sebastian T B, Klein P N, Kimia B B. On aligning curves[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, 25(1): 116-125. [DOI:10.1109/TPAMI.2003.1159951]
- [6] 黄蕾. 基于细胞神经网络的图像边缘提取算法研究. 计算机软件及计算机应用, 中国知网, 2006年1月1日.
- [7] 单紫薇. 基于机器视觉测量的齿轮图像边界提取算法研究. 机械工业, 中国知网, 2017年6月2日.
- [8] 田建华. 基于灰度信息和 Harris 角点检测的眼睛特征提取. 计算机软件及计算机应用, 新课程（教育学术）, 2011年04期.
- [9] 姬周强. 基于轮廓特征的二维碎片拼合技术研究. 计算机软件及计算机应用, 中国知网, 2009年4月1日.
- [10] 危辉, 李俐. 拱序列的曲线描述与匹配. 中国图像图形学报, 2017年08期.
- [11] 郑一露. 基于字符串编辑距离的图像检索算法. 信息通信, 2015年07期.
- [12] 张伟东. 基于遗传算法与动态规划法混合算法的铁路纵断面优化设计. 铁路运输, 中国知网, 2015年4月1日.

附录 1

```

A=imread('C:\Users\hello\Desktop\2.jpg');
A=rgb2gray(A);
imshow(A);
[a,b]=size(A);
T=zeros(a,b);
T1=zeros(a,b);
T2=zeros(a,b);
T3=zeros(a,b);
T4=zeros(a,b);
n=70;
for p=2:a-1
for q=2:b-1
if (A(p,q)-A(p,q+1))>n| (A(p,q)-A(p,q-1))>n| (A(p,q)-A(p+1,q))>n|
(A(p,q)-A(p-1,q))>n| (A(p,q)-A(p-1,q+1))>n| (A(p,q)-A(p+1,q-1))>n|
(A(p,q)-A(p-1,q-1))>n| (A(p,q)-A(p+1,q+1))>n
T1(p,q)=255;break;
end
end
end
figure;imshow(T1);
for q=2:b-1
for p=2:a-1
if (A(p,q)-A(p,q+1))>n| (A(p,q)-A(p,q-1))>n| (A(p,q)-A(p+1,q))>n|
(A(p,q)-A(p-1,q))>n| (A(p,q)-A(p-1,q+1))>n| (A(p,q)-A(p+1,q-1))>n|
(A(p,q)-A(p-1,q-1))>n| (A(p,q)-A(p+1,q+1))>n
T2(p,q)=255;break;
end
end
end
figure;imshow(T2);
for p=a-1:-1:2
for q=b-1:-1:2
if (A(p,q)-A(p,q+1))>n| (A(p,q)-A(p,q-1))>n| (A(p,q)-A(p+1,q))>n|

```

```

(A(p,q)-A(p-1,q))>n| (A(p,q)-A(p-1,q+1))>n| (A(p,q)-A(p+1,q-1))>n|
(A(p,q)-A(p-1,q-1))>n| (A(p,q)-A(p+1,q+1))>n
T3(p,q)=255;break;
end
end
end
figure;imshow(T3);
for q=b-1:-1:2
for p=a-1:-1:2
if (A(p,q)-A(p,q+1))>n| (A(p,q)-A(p,q-1))>n| (A(p,q)-A(p+1,q))>n|
(A(p,q)-A(p-1,q))>n| (A(p,q)-A(p-1,q+1))>n| (A(p,q)-A(p+1,q-1))>n|
(A(p,q)-A(p-1,q-1))>n| (A(p,q)-A(p+1,q+1))>n
T4(p,q)=255;break;
end
end
end
figure;imshow(T4);
T=T1+T2+T3+T4;
figure;imshow(T);

```

附录 2

```

A=imread('C:\Users\hello\Desktop\2.jpg');
A=rgb2gray(A);
A=im2bw(A,0.5);
imshow(A);
[a,b]=size(A);
B=ones(a,b);
for m=1:a
for n=1:b
if(A(m,n)==0)
    B(m,n)=0;
    startx=m;
    starty=n;

```

```

    return;
end
end
end
findstartpoint=1;
startDirect=1;
findpoint=1;
x=startx;
y=starty;
direction={{0,-1},{-1,-1},{-1,0},{-1,1},{0,1},{1,1},{1,0},{1,-1}};
while findstartpoint
    findpoint=1;
    while findpoint
        nextx=x+direction{startDirect}{1};
        nexty=y+direction{startDirect}{2};
        v=A(nextx,nexty);
        if(v)
            startDirect=startDirect+1;
            if(startDirect>8)
                startDirect=startDirect-8;
            else
                startDirect=startDirect;
            end
        else
            B(nextx,nexty)=0;
            findpoint=0;
            x=nextx;
            y=nexty;
            if x==startx & y==starty
                findstartpoint=0;
            else
                startDirect=startDirect-2;
                if(startDirect<1)
                    startDirect=startDirect+8;
                else

```

```

startDirect=startDirect;
    end
    end
    end
    end
end
figure;imshow(B);

```

附录 3

```

A=imread('C:\Users\hello\Desktop\2.jpg');
A=rgb2gray(A);
A=im2bw(A,0.5);
[a,b]=size(A);
B=ones(a,b);
for m=1:a
for n=1:b
if(A(m,n)==0)
    B(m,n)=0;
    startx=m;
    starty=n;
    A2{1}{1}=m;
    A2{1}{2}=n;
    return;
end
end
end
findstartpoint=1;
startDirect=1;
findpoint=1;
n2=1;
x=startx;
y=starty;
direction={0,-1},{-1,-1},{-1,0},{-1,1},{0,1},{1,1},{1,0},{1,-1}};
while findstartpoint
    findpoint=1;
    while findpoint
        nextx=x+direction{startDirect}{1};
        nexty=y+direction{startDirect}{2};
        v=A(nextx,nexty);
        if(v)

```

```

startDirect=startDirect+1;
if(startDirect>8)
    startDirect=startDirect-8;
else
    startDirect=startDirect;
end
else
    B(nextx,nexty)=0;

    findpoint=0;
    x=nextx;
    y=nexty;
    if x==startx & y==starty
        findstartpoint=0;
    else
        n2=n2+1;
        A2{n2}{1}=nextx;
        A2{n2}{2}=nexty;
        startDirect=startDirect-2;
        if(startDirect<1)
            startDirect=startDirect+8;
        else
            startDirect=startDirect;
        end
    end
end
end
end
end
fx=[-2 -1 0 1 2];
lx=filter2(fx,A);
fy=[-2;-1;0;1;2];
ly=filter2(fy,A);
lx2=lx.^2;
ly2=ly.^2;
lxy=lx.*ly;
clear lx;
clear ly;
h=fspecial('gaussian',[7 7],2);
lx2=filter2(h,lx2);
ly2=filter2(h,ly2);
lxy=filter2(h,lxy);
height=size(B,1);
width=size(B,2);
result=zeros(height,width);

```

```

R=zeros(height,width);
Rmax=0;
for i=1:height
for j=1:width
M=[lx2(i,j) lxy(i,j);lxy(i,j) ly2(i,j)];
R(i,j)=det(M)-0.08*(trace(M)^2);
    if R(i,j)>Rmax
        Rmax=R(i,j);
    end
end
end
cnt=0;
n3=0;
for i=2:height-1
for j=2:width-1
    if R(i,j)>0.01*Rmax && R(i,j)>R(i-1,j-1) && R(i,j)>R(i-1,j) && R(i,j)>R(i-1,j+1)
    && R(i,j)>R(i,j-1) && R(i,j)>R(i,j+1) && R(i,j)>R(i+1,j-1) && R(i,j)>R(i+1,j) &&
    R(i,j)>R(i+1,j+1)
        result(i,j)=1;
        if B(i,j)==0
            result(i,j)=result(i,j);
            cnt=cnt+1;
            n3=n3+1;
            A3{n3}{1}=i;
            A3{n3}{2}=j;
        else
            result(i,j)=0;
        end
    end
end
end
end
[posc,posr]=find(result==1);
cnt
figure;imshow(B);
hold on;
plot(posr,posc,'r');

```

附录 4

```

A=imread('C:\Users\hello\Desktop\7.jpg');
%A=rgb2gray(A);
A=im2bw(A,0.5);
[a,b]=size(A);

```

```

B=ones(a,b);
for m=1:a
for n=1:b
if(A(m,n)==0)
    B(m,n)=0;
    startx=m;
    starty=n;
    A2{1}{1}=m;
    A2{1}{2}=n;
    return;
end
end
end
findstartpoint=1;
startDirect=1;
findpoint=1;
n2=1;
x=startx;
y=starty;
direction={ {0,-1},{-1,-1},{-1,0},{-1,1},{0,1},{1,1},{1,0},{1,-1}};
while findstartpoint
    findpoint=1;
    while findpoint
        nextx=x+direction{startDirect}{1};
        nexty=y+direction{startDirect}{2};
        v=A(nextx,nexty);
        if(v)
            startDirect=startDirect+1;
            if(startDirect>8)
                startDirect=startDirect-8;
            else
                startDirect=startDirect;
            end
        else
            B(nextx,nexty)=0;

            findpoint=0;
            x=nextx;
            y=nexty;
            if x==startx & y==starty
                findstartpoint=0;
            else
                n2=n2+1;
                A2{n2}{1}=nextx;

```



```

        A2{n2}{2}=nexty;
        startDirect=startDirect-2;
        if(startDirect<1)
            startDirect=startDirect+8;
        else
            startDirect=startDirect;
        end
    end
end
end
end
K1=ones(a,b);
cnt=0;
for t7=n2+1:n2*2-10
    A2{t7}{1}=A2{t7-n2}{1};
    A2{t7}{2}=A2{t7-n2}{2};
end
for r7=2:n2+1
    if A2{r7+1}{1}==A2{r7-1}{1}
        A9=1;
        B9=0;
        C9=-1*A2{r7+1}{1};
    else
        A9=(A2{r7+1}{2}-A2{r7-1}{2})/(A2{r7+1}{1}-A2{r7-1}{1});
        B9=-1;
        C9=A2{r7-1}{2}-A9*A2{r7-1}{1};
    end
    h6=abs(A9*A2{r7}{1}+B9*A2{r7}{2}+C9)/((A9^2+B9^2)^0.5);
    l5=(((A2{r7}{2}-A2{r7-1}{2})^2)+((A2{r7}{1}-A2{r7-1}{1})^2))^0.5;
    l6=(((A2{r7+1}{2}-A2{r7}{2})^2)+((A2{r7+1}{1}-A2{r7}{1})^2))^0.5;
    jiajiao=acos(h6/l5)+acos(h6/l6);
    if jiajiao<pi
        cnt=cnt+1;
        A3{cnt}{1}=A2{r7}{1};
        A3{cnt}{2}=A2{r7}{2};
        K1(A3{cnt}{1},A3{cnt}{2})=0;
    end
end
cnt2=0;
A3{cnt+1}{1}=A3{1}{1};
A3{cnt+1}{2}=A3{1}{2};
A3{cnt+2}{1}=A3{2}{1};
A3{cnt+2}{2}=A3{2}{2};
A3{cnt+3}{1}=A3{3}{1};

```

```

A3{cnt+3}{2}=A3{3}{2};
A3{cnt+4}{1}=A3{4}{1};
A3{cnt+4}{2}=A3{4}{2};
A3{cnt+5}{1}=A3{5}{1};
A3{cnt+5}{2}=A3{5}{2};
A3{cnt+6}{1}=A3{6}{1};
A3{cnt+6}{2}=A3{6}{2};
K=ones(a,b);
for r8=4:cnt+3
    A10=(A3{r8+3}{2}-A3{r8-3}{2})/(A3{r8+3}{1}-A3{r8-3}{1});
    B10=-1;
    C10=A3{r8-3}{2}-A10*A3{r8-3}{1};
    h16=abs(A10*A3{r8}{1}+B10*A3{r8}{2}+C10)/((A10^2+B10^2)^0.5);
    l15=(((A3{r8}{2}-A3{r8-3}{2})^2)+((A3{r8}{1}-A3{r8-3}{1})^2))^0.5;
    l16=(((A3{r8+3}{2}-A3{r8}{2})^2)+((A3{r8+3}{1}-A3{r8}{1})^2))^0.5;
    jiajiao1=acos(h16/l15)+acos(h16/l16);
    if jiajiao1<((5*pi)/6)
        cnt2=cnt2+1;
        A13{cnt2}{1}=A3{r8}{1};
        A13{cnt2}{2}=A3{r8}{2};
        K(A13{cnt2}{1},A13{cnt2}{2})=0;
    end
end
[posr,posc]=find(K==0);
figure;imshow(B);
hold on;
plot(posc,posr,'r');

```

附录 5

```

A=imread('C:\Users\hello\Desktop\7.jpg');
%A=rgb2gray(A);
A=im2bw(A,0.5);
[a,b]=size(A);
B=ones(a,b);
for m=1:a
    for n=1:b
        if(A(m,n)==0)
            B(m,n)=0;
            startx=m;
            starty=n;
            A2{1}{1}=m;
            A2{1}{2}=n;

```

```

    return;
end
end
end
findstartpoint=1;
startDirect=1;
findpoint=1;
n2=1;
x=startx;
y=starty;
direction={{0,-1},{-1,-1},{-1,0},{-1,1},{0,1},{1,1},{1,0},{1,-1}};
while findstartpoint
    findpoint=1;
    while findpoint
        nextx=x+direction{startDirect}{1};
        nexty=y+direction{startDirect}{2};
        v=A(nextx,nexty);
        if(v)
            startDirect=startDirect+1;
            if(startDirect>8)
                startDirect=startDirect-8;
            else
                startDirect=startDirect;
            end
        else
            B(nextx,nexty)=0;
            findpoint=0;
            x=nextx;
            y=nexty;
            if x==startx & y==starty
                findstartpoint=0;
            else
                n2=n2+1;
                A2{n2}{1}=nextx;
                A2{n2}{2}=nexty;
                startDirect=startDirect-2;
                if(startDirect<1)
                    startDirect=startDirect+8;
                else
                    startDirect=startDirect;
                end
            end
        end
    end
end
end
end

```

```

end
K1=ones(a,b);
cnt=0;
for t7=n2+1:n2*2-10
    A2{t7}{1}=A2{t7-n2}{1};
    A2{t7}{2}=A2{t7-n2}{2};
end
for r7=2:n2+1
    if A2{r7+1}{1}==A2{r7-1}{1}
        A9=1;
        B9=0;
        C9=-1*A2{r7+1}{1};
    else
        A9=(A2{r7+1}{2}-A2{r7-1}{2})/(A2{r7+1}{1}-A2{r7-1}{1});
        B9=-1;
        C9=A2{r7-1}{2}-A9*A2{r7-1}{1};
    end
    h6=abs(A9*A2{r7}{1}+B9*A2{r7}{2}+C9)/((A9^2+B9^2)^0.5);
    l5=((A2{r7}{2}-A2{r7-1}{2})^2)+((A2{r7}{1}-A2{r7-1}{1})^2))^0.5;
    l6=((A2{r7+1}{2}-A2{r7}{2})^2)+((A2{r7+1}{1}-A2{r7}{1})^2))^0.5;
    jiajiao=acos(h6/l5)+acos(h6/l6);
    if jiajiao<pi
        cnt=cnt+1;
        A3{cnt}{1}=A2{r7}{1};
        A3{cnt}{2}=A2{r7}{2};
        K1(A3{cnt}{1},A3{cnt}{2})=0;
    end
end
cnt2=0;
A3{cnt+1}{1}=A3{1}{1};
A3{cnt+1}{2}=A3{1}{2};
A3{cnt+2}{1}=A3{2}{1};
A3{cnt+2}{2}=A3{2}{2};
A3{cnt+3}{1}=A3{3}{1};
A3{cnt+3}{2}=A3{3}{2};
A3{cnt+4}{1}=A3{4}{1};
A3{cnt+4}{2}=A3{4}{2};
A3{cnt+5}{1}=A3{5}{1};
A3{cnt+5}{2}=A3{5}{2};
A3{cnt+6}{1}=A3{6}{1};
A3{cnt+6}{2}=A3{6}{2};
K=ones(a,b);
for r8=4:cnt+3
    A10=(A3{r8+3}{2}-A3{r8-3}{2})/(A3{r8+3}{1}-A3{r8-3}{1});

```

```

B10=-1;
C10=A3{r8-3}{2}-A10*A3{r8-3}{1};
h16=abs(A10*A3{r8}{1}+B10*A3{r8}{2}+C10)/((A10^2+B10^2)^0.5);
l15=(((A3{r8}{2}-A3{r8-3}{2})^2)+((A3{r8}{1}-A3{r8-3}{1})^2))^0.5;
l16=(((A3{r8+3}{2}-A3{r8}{2})^2)+((A3{r8+3}{1}-A3{r8}{1})^2))^0.5;
jiajiao1=acos(h16/l15)+acos(h16/l16);
if jiajiao1<((5*pi)/6)
    cnt2=cnt2+1;
    A13{cnt2}{1}=A3{r8}{1};
    A13{cnt2}{2}=A3{r8}{2};
    K(A13{cnt2}{1},A13{cnt2}{2})=0;
end
End
[posr,posc]=find(K==0);
figure;imshow(B);
hold on;
plot(posc,posr,'r');
c2=0;
for i2=1:n2
    for j2=1:cnt2
        if A2{i2}{1}=A13{j2}{1} && A2{i2}{2}=A13{j2}{2}
            c2=c2+1;
            A6{c2}{1}=A2{i2}{1};
            A6{c2}{2}=A2{i2}{2};
            C{c2}{1}=i2;
        end
    end
end
C{cnt2+1}{1}=C{1}{1}+n2;
C{cnt2+2}{1}=C{2}{1}+n2;
A6{c2+1}{1}=A6{1}{1};
A6{c2+1}{2}=A6{1}{2};
A6{c2+2}{1}=A6{2}{1};
A6{c2+2}{2}=A6{2}{2};
a5=0;
for p=1:c2
    x1=A6{p}{1};
    y1=A6{p}{2};
    x2=A6{p+2}{1};
    y2=A6{p+2}{2};
    l=(((x2-x1)^2)+((y2-y1)^2))^0.5;
    A4=(y2-y1)/(x2-x1);
    B4=-1;
    C4=y1-A4*x1;

```

```

qmax=-1;
d=0;
s=0;
for q=C{p}{1}:C{p+2}{1}
    if q>n2
        q=q-n2;
    end
    q1=abs(A4*A2{q}{1}+B4*A2{q}{2}+C4)/((A4^2+B4^2)^0.5);
    if q1>qmax
        qmax=q1;
        x3=A2{q}{1};
        y3=A2{q}{2};
    end
    h10=qmax;
    s=((A2{q}{1}-A2{q+1}{1})^2+(A2{q}{2}-A2{q+1}{2})^2)^(1/2)+s;
end
x4=(y3-y1-A4*x3+A4*x1)/(2*A4);
y4=A4*x4+y1-A4*x1;
x5=(x3+x4)/2;
y5=(y3+y4)/2;
for Q=C{p}{1}:C{p+2}{1}
    q2=A4*A2{Q}{1}+y5-A4*x5-A2{Q}{2};
    if q2<=0.5
        d=d+1;
        D{d}{1}=A2{Q}{1};
        D{d}{2}=A2{Q}{2};
    end
end
a10=h10/l;
v10=s/l;
b10=((((D{1}{1}-D{2}{1})^2)+((D{1}{2}-D{2}{2})^2))^(1/2))/l;
o10=h10/(((x3-(x1+x2)/2)^2+(y3-(y1+y2)/2)^2)^(1/2));
a5=a5+1;
E{a5}={a10,b10,v10,o10};
end
A=imread('C:\Users\hello\Desktop\5.jpg');
%A=rgb2gray(A);
A=im2bw(A,0.5);
[a,b]=size(A);
B=ones(a,b);
for m=1:a
    for n=1:b
        if(A(m,n)==0)
            B(m,n)=0;

```

```

    startx=m;
    starty=n;
    A2{1}{1}=m;
    A2{1}{2}=n;
    return;
end
end
end
findstartpoint=1;
startDirect=1;
findpoint=1;
n2=1;
x=startx;
y=starty;
direction={{0,-1},{-1,-1},{-1,0},{-1,1},{0,1},{1,1},{1,0},{1,-1}};
while findstartpoint
    findpoint=1;
    while findpoint
        nextx=x+direction{startDirect}{1};
        nexty=y+direction{startDirect}{2};
        v=A(nextx,nexty);
        if(v)
            startDirect=startDirect+1;
            if(startDirect>8)
                startDirect=startDirect-8;
            else
                startDirect=startDirect;
            end
        else
            B(nextx,nexty)=0;
            findpoint=0;
            x=nextx;
            y=nexty;
            if x==startx & y==starty
                findstartpoint=0;
            else
                n2=n2+1;
                A2{n2}{1}=nextx;
                A2{n2}{2}=nexty;
                startDirect=startDirect-2;
                if(startDirect<1)
                    startDirect=startDirect+8;
                else
                    startDirect=startDirect;
                end
            end
        end
    end
end

```

```

        end
    end
end
end
end
K1=ones(a,b);
cnt=0;
for t7=n2+1:n2*2-10
    A2{t7}{1}=A2{t7-n2}{1};
    A2{t7}{2}=A2{t7-n2}{2};
end
for r7=2:n2+1
    if A2{r7+1}{1}==A2{r7-1}{1}
        A9=1;
        B9=0;
        C9=-1*A2{r7+1}{1};
    else
        A9=(A2{r7+1}{2}-A2{r7-1}{2})/(A2{r7+1}{1}-A2{r7-1}{1});
        B9=-1;
        C9=A2{r7-1}{2}-A9*A2{r7-1}{1};
    end
    h6=abs(A9*A2{r7}{1}+B9*A2{r7}{2}+C9)/((A9^2+B9^2)^0.5);
    l5=(((A2{r7}{2}-A2{r7-1}{2})^2)+((A2{r7}{1}-A2{r7-1}{1})^2))^0.5;
    l6=(((A2{r7+1}{2}-A2{r7}{2})^2)+((A2{r7+1}{1}-A2{r7}{1})^2))^0.5;
    jiajiao=acos(h6/l5)+acos(h6/l6);
    if jiajiao<pi
        cnt=cnt+1;
        A3{cnt}{1}=A2{r7}{1};
        A3{cnt}{2}=A2{r7}{2};
        K1(A3{cnt}{1},A3{cnt}{2})=0;
    end
end
cnt2=0;
A3{cnt+1}{1}=A3{1}{1};
A3{cnt+1}{2}=A3{1}{2};
A3{cnt+2}{1}=A3{2}{1};
A3{cnt+2}{2}=A3{2}{2};
A3{cnt+3}{1}=A3{3}{1};
A3{cnt+3}{2}=A3{3}{2};
A3{cnt+4}{1}=A3{4}{1};
A3{cnt+4}{2}=A3{4}{2};
A3{cnt+5}{1}=A3{5}{1};
A3{cnt+5}{2}=A3{5}{2};
A3{cnt+6}{1}=A3{6}{1};

```



```

A3{cnt+6}{2}=A3{6}{2};
K=ones(a,b);
for r8=4:cnt+3
    A10=(A3{r8+3}{2}-A3{r8-3}{2})/(A3{r8+3}{1}-A3{r8-3}{1});
    B10=-1;
    C10=A3{r8-3}{2}-A10*A3{r8-3}{1};
    h16=abs(A10*A3{r8}{1}+B10*A3{r8}{2}+C10)/((A10^2+B10^2)^0.5);
    l15=(((A3{r8}{2}-A3{r8-3}{2})^2)+((A3{r8}{1}-A3{r8-3}{1})^2))^0.5;
    l16=(((A3{r8+3}{2}-A3{r8}{2})^2)+((A3{r8+3}{1}-A3{r8}{1})^2))^0.5;
    jiajiao1=acos(h16/l15)+acos(h16/l16);
    if jiajiao1<((5*pi)/6)
        cnt2=cnt2+1;
        A13{cnt2}{1}=A3{r8}{1};
        A13{cnt2}{2}=A3{r8}{2};
        K(A13{cnt2}{1},A13{cnt2}{2})=0;
    end
end
[posr,posc]=find(K==0);
figure;imshow(B);
hold on;
plot(posc,posr,'r');
c2=0;
for i2=1:n2
    for j2=1:cnt2
        if A2{i2}{1}==A13{j2}{1} && A2{i2}{2}==A13{j2}{2}
            c2=c2+1;
            A6{c2}{1}=A2{i2}{1};
            A6{c2}{2}=A2{i2}{2};
            C{c2}{1}=i2;
        end
    end
end
C{cnt2+1}{1}=C{1}{1}+n2;
C{cnt2+2}{1}=C{2}{1}+n2;
A6{c2+1}{1}=A6{1}{1};
A6{c2+1}{2}=A6{1}{2};
A6{c2+2}{1}=A6{2}{1};
A6{c2+2}{2}=A6{2}{2};
a15=0;
for p=1:c2
    x1=A6{p}{1};
    y1=A6{p}{2};
    x2=A6{p+2}{1};
    y2=A6{p+2}{2};

```

```

l=(((x2-x1)^2)+((y2-y1)^2))^0.5;
A4=(y2-y1)/(x2-x1);
B4=-1;
C4=y1-A4*x1;
qmax=-1;
d=0;
s=0;
for q=C{p}{1}:C{p+2}{1}
    if q>n2
        q=q-n2;
    end
    q1=abs(A4*A2{q}{1}+B4*A2{q}{2}+C4)/((A4^2+B4^2)^0.5);
    if q1>qmax
        qmax=q1;
        x3=A2{q}{1};
        y3=A2{q}{2};
    end
    h10=qmax;
    s=((A2{q}{1}-A2{q+1}{1})^2+(A2{q}{2}-A2{q+1}{2})^2)^(1/2)+s;
end
x4=(y3-y1-A4*x3+A4*x1)/(2*A4);
y4=A4*x4+y1-A4*x1;
x5=(x3+x4)/2;
y5=(y3+y4)/2;
for Q=C{p}{1}:C{p+2}{1}
    q2=A4*A2{Q}{1}+y5-A4*x5-A2{Q}{2};
    if q2<=0.5
        d=d+1;
        D{d}{1}=A2{Q}{1};
        D{d}{2}=A2{Q}{2};
    end
end
a10=h10/l;
v10=s/l;
b10=(((D{1}{1}-D{2}{1})^2)+((D{1}{2}-D{2}{2})^2))^(1/2)/l;
o10=h10/(((x3-(x1+x2)/2)^2+(y3-(y1+y2)/2)^2)^(1/2));
a15=a15+1;
E2{a15}={a10,b10,v10,o10};
end
n6=zeros(a5,a15);
D6=zeros(a5,a15);
for i4=1:a5
    for j4=1:a15
        D6(i4,j4)=abs(E{i4}{1}-E2{j4}{1})/min(E{i4}{1},E2{j4}{1})+abs(E{i4}{2}-E2{j4}{2})
    end
end

```

```

{2})/min(E{i4}{2},E2{j4}{2}))+abs(E{i4}{3}-E2{j4}{3})/min(E{i4}{3},E2{j4}{3})
+abs(E{i4}{4}/(E{i4}{4}+1)-E{j4}{4}/(E{j4}{4}+1)));
if i4==1 & j4==1
    n6(i4,j4)=0;
else if i4==1 & j4>1
    n6(i4,j4)=n6(i4,j4-1)+D6(i4,j4);
else if i4>1 & j4==1
    n6(i4,j4)=n6(i4-1,j4)+D6(i4,j4);
else if i4>1 & j4>1
    n6(i4,j4)=min(min(n6(i4-1,j4-1)+D6(i4,j4),n6(i4-1,j4)+D6(i4,j4)),n6(i
4,j4-1)+D6(i4,j4));
    end
end
end
end
n6(a5,a15)

```