

摘 要

随着信息化的普及和办公自动化程度的不断提高,传统的印章管理手段已经无法满足现代业务的需求,包括印章的存档和鉴别。特别是在公安、金融领域,更加迫切需要利用信息技术进行印章识别的系统,而特征提取是印章识别系统的最主要组成部分。因此,本文以印章图像为研究对象,主要研究印章的特征提取。

论文首先阐述了目前印章识别系统的需求和国内外的研究现状,介绍了当前印章识别系统的基本结构,各部分的主要任务以及面临的问题。接着按照印章特征提取的过程详细介绍相应的方法,包括印章图像颜色特征的提取、高准确度的二值化、图像去噪等预处理方法,结合当前的轮廓提取算法的印章形状特征和几何特征提取方法,以及几何特征的精确计算方法。论文最后介绍了基于字符定位和图像分割算法的字符区域特征的提取方法。

在预处理过程中,最重要的步骤是图像的二值化。在研究了几种有代表性的算法之后,论文采用了能突出字符细节特征的 LOG 算子进行二值化处理。

在印章形状的识别过程中,论文提出了结合链码跟踪法和余弦法的轮廓端点提取算法,并进一步根据端点的个数和几何位置确定方形、圆形、椭圆形和三角形印章的形状。

此外,针对圆形或椭圆形印章,论文提出了一种基于链码跟踪的形心校正算法,解决了由于形心定位误差导致字符区域无法定位的情况。还提出了一种基于改进 Bresenham 算法的自适应极坐标变换方法,利用该算法对圆形和椭圆形印章进行极坐标变换,然后定位印章中字符区域的位置。

实验表明上述算法能够有效地对印章的形状、几何、字符区域等特征进行提取。

关键词 印章, 印章识别, 特征提取, 自适应的极坐标变换方法

Abstract

With unceasing enhancement of the office automation degree, traditional manual recognition seal method is now unable to satisfy the need of massive services. Thus, Seal Recognition Systems based on Information Technology are now the needing systems of the finance and public security department. Feature extraction is the most important part of Seal Recognition Systems. Thus, this dissertation mainly presents researches on feature extraction of seal images.

This dissertation first introduces the requirement on Seal Image Recognition Systems. It also presents the main architecture of the main Seal Image Recognition Systems and the difficulty in the recognition process. Then, this dissertation introduces methods in the feature extraction processes. It shows details of the section of image preprocessing. This section includes the color feature extraction, high quality binarization and noise removal of the color images. Then, based on image preprocessing, the paper shows the method of geometry feature extraction. Finally, it proposes the method of the character part segmentation.

In the preprocessing process, the most important step is the image binarization. After studying several representative algorithms, we use the LOG arithmetic operator method to do binarization which can enhance the edge data of characters .

We use the chain code track algorithm to withdraw the outline of the seal, and through rejecting the smooth spots in the outline to obtain the vertices of the seal. Then, through the number and geometry position of the vertices, we obtain the shape of rectangular, circular, elliptic and triangular seals.

It is clear that if we can't find the right center point of circular or elliptic seals, then we can't extract the character part feature successfully. So we come out an inner contour extraction algorithm to rectify the geometry center of the seals. We also come out an adaptive polar coordinate algorithm based on Bresenham algorithm to transform the seal to polar coordinate.

It is proved that those algorithms can do shape extraction, geometry feature extraction and character part feature extraction effectively.

Keywords Seal, Seal Image Recognition , Feature Extraction, Adaptive Polar Coordinate Algorithm

图目录

图 1-1 印章自动识别系统结构示意图.....	2
图 1-2 印章预处理模块.....	3
图 1-3 本文的研究路线.....	5
图 2-1 Ostu 算法流程图.....	11
图 2-2 处理样章.....	12
图 2-3 $\nabla^2 G$ 的截面图, 其中 W 为主瓣宽度.....	14
图 2-4 基于 LOG 算子的局部动态阈值算法.....	15
图 2-5 双峰直方图阈值分割法(a)和 Laplacian 算子法(b)的二值化效果.....	16
图 2-6 基于 LOG 算子的局部动态阈值算法的二值化效果.....	16
图 2-7 论文中使用的去噪算法伪代码.....	18
图 2-8 去噪后的效果图.....	18
图 3-1 基于扫描线的轮廓提取算法示意图.....	21
图 3-2 链码跟踪提取轮廓算法.....	22
图 3-3 外边缘有断裂的印章的轮廓提取结果.....	24
图 4-1 计算几何中心算法.....	26
图 4-2 找到错误的几何中心.....	27
图 4-3 计算轮廓点到形心的距离最大和最小值算法.....	27
图 4-4 计算圆形与椭圆形印章几何参数算法流程.....	28
图 4-5 一条与原点距离为 s , 方向角为 θ 的直线.....	29
图 4-6 过顶点 A 与边缘夹角为 θ 的直线.....	31
图 4-7 二值化图像反色所得图像.....	32
图 4-8 精确定位后的圆心位置.....	32
图 4-9 椭圆倾斜校正前和校正后的结果.....	33
图 5-1 椭圆中心矩弦长示意图.....	36
图 5-2 圆形的自适应的极坐标变换算法.....	41
图 5-3 一般极坐标变换和自适应极坐标变换方法的结果对比.....	42
图 5-4 环形投影结果.....	42
图 5-5 字符区域中心射线方向投影结果.....	43
图 5-6 椭圆的极坐标变换方法.....	45

表目录

表 3-1 印章形状特征提取结果.....	24
表 5-1 圆形字符分割角度结果.....	43

第1章 绪论

1.1 课题背景

古往今来,印章在东方国家作为一种标识诚信的事物在各种场合中发挥着重要作用。21世纪以来,随着信息化、计算机技术的发展,虽然出现了诸如电子签名、电子印章等含有高级防伪技术的新的信证凭证,然而印章作为符合人们传统习惯和具有法律效力的文档标记的作用依然存在,仍是当前的主要信证凭证。

但是,由于仿制技术的不断提高,印章被仿制和造假的案例时有发生,已经造成了很大的经济损失和严重的社会影响。因此,近年来国家出台了很多规定和标准来规范印章尤其是公章的制作和管理工作。早在1999年10月31日,国务院就印发了《国务院关于国家行政机关和企业事业单位社会团体印章管理的规定》(国发[1999]25号)文件(以下简称《规定》),该《规定》将国家行政机关和企业事业单位、社会团体的行政公章和其他专用印章都纳入了其管理范畴,规范这些印章的生产和运作。2000年3月18日,公安部发布了《中华人民共和国公共安全行业标准(印章治安管理信息系统)》{GA 241.1~241.8—2000},该标准于2000年4月1日起正式实施。2004年1月17日,公安部又发布了(公治办[2004]40号)文件,规定“凡冠以单位名称的专用印章均属公章”,“应依据《规定》及公安部三局1991年《关于企业单位刻制公章问题的批复》”,都应该纳入公章管理范围。上述这些规定和标准对于当前国家机关和企业事业单位印章的管理工作起到了非常重要的作用。

尽管如此,在防伪程度较高的电子签名、电子印章尚未完全普及的今天,利用伪造的印章进行犯罪活动的案件依然层出不穷,这就需要研究更加有效的方法来管理印章,包括印章的存储和识别。目前,印章识别和管理系统的应用越来越广,特别在金融和公安领域。例如,银行根据存储备案的印章图像或提取的印章特征信息,对支票上的印章进行真伪鉴别。公安部门在侦破伪造印章案件时,对提取到的印章图像进行原型比对,或提取特征以便于辨别真伪等。

在印章识别系统中,如何提取印章的特征是其中的核心技术之一。现在很多研究机构都在研究印章的特征提取方法,以利于管理和识别印章。印章特征提取成为当前印章识别研究的重点。随着办公自动化程度的提高,可以想象到,将来印章识别系统将像打印机一样可能成为办公场所的必备工具。因此,对印章识别特别是特征提取技术的研究,以及相关应用系统的开发具有重要的现实意义和实

用价值，以及广阔的市场和巨大的商业应用前景。

1.2 国内外研究现状

印章识别系统已经被广泛用于公安、金融、办公场所等各个领域。由于市场的需求相当大，现在国内相当多的研究机构都开发了各自的印章识别系统，这些可以分成两大类，一类是通用性系统，例如深圳大学的张小棉等人^[1-2]开发的印章管理系统，该系统非常庞大，集面向对象的印章排版、制造印章、管理印章、自动识别印章、匹配印章等功能于一身。该系统的设计采用客户端/服务器的结构，并且将面向对象的技术引入印章的管理和识别之中；另一类是针对某个行业的专用系统，如清华大学的周琨等^[3]研制的金融行业印章自动识别系统。长沙睿思科技开发有限公司^[31]开发的面向银行的印章识别组件系统等。

典型印章自动鉴别系统的框架如图 1-1 所示。系统通过终端的扫描仪等工具采集真彩色印章图像，并且从原始印章数据库中提取对应的原始印章图像。接着通过预处理模块对图像进行处理，然后送入图像匹配模块，在该模块里，提取输入图像的特征并与原始印章进行匹配，识别结果送入后处理器，在这里进行自动校验或者人工校验，最后输出结果。

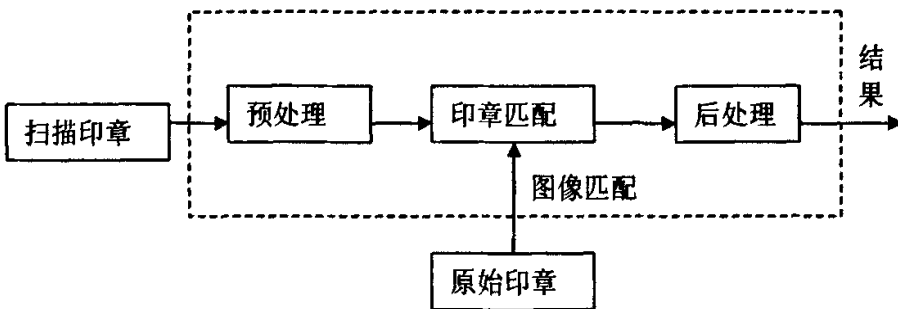


图 1-1 印章自动识别系统结构示意图

由于印章盖印条件的差异，包括票据背景的复杂程度，盖印的用力大小，印泥的多少等都会对印章图像产生极大的影响，因此，就算是同一印章得到的图像也很难一致。在这种情况下，就需要系统具有鲁棒性。否则如果出现将伪章识别成为有效印章的情况，可能会出现重大案件，给社会造成很大损失和恶劣的影响。系统还必须兼顾实时性和可操作性，以适应网络交互和大面积推广。

从图 1-1 可以看出，印章的自动识别系统最主要的三个模块分别是：预处理模块^[4]、印章匹配^[5-6]模块和后处理模块^[3]。预处理模块的目的是将图像处理成适

宜于识别的二值低噪声图像。印章匹配模块中最主要的是对输入印章图片的特征提取。而后处理模块^[3]则是根据匹配的结果确定边缘误差、像素点误差、面积误差、局部误差等等以确定印章是否与原印章匹配^[7-8]。下面一节中，我们将对本文所关心的印章图像预处理和特征提取的现状进行阐述。

1.2.1 预处理技术研究现状

扫描到印章图像一般可分为三个部分：白色或者灰色的背景，黑色的文字如手写签名、日期等噪声，以及呈红色或者蓝色的目标区域。预处理的目的是将目标印章从图像中提取出来，得到清晰、二值化、低噪声的印章图像。预处理的步骤如图 1-2 所示。

首先利用彩色信息进行图像分割，确定印章图像中红色或者蓝色光的 RGB 阈值。一种分割方法是根据采集的大量样本或者统计值的最大值来确定最佳阈值。另一种分割方法是将 RGB 空间映射到 YUV 或者 HIS 空间，从而求得最佳值。有的印章识别系统还增加了人工干预来解决个别无法识别的样本问题^[2]。

由于提取出来的印章图像可能因为印泥的不均匀、噪声过多而出现残缺、断裂等问题，因而第二步是对图像进行增强和去除背景噪声点处理。接着对提取出来的印章图像进行二值化处理。不同的印章识别系统使用的二值化算法有所不同。我们将在第二章进行研究，最后是对二值化图像进行去噪处理。

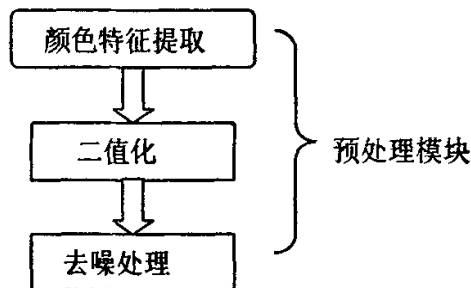


图 1-2 印章预处理模块

1.2.2 特征提取研究现状

1.2.2.1 形状特征提取

鉴于印章图像的外轮廓都是凸多边形而且是连续的，可以通过行列扫描方式^[9]确定图形的外轮廓。从所得到的外轮廓向量取得顶点坐标。根据顶点坐标的几

何位置组合特征可以确定印章图像的形状特征。系统中可鉴别的印章形状有：方形、矩形、圆形、椭圆、菱形、三角形等。

1.2.2.2 几何特征提取

根据顶端点坐标的值和组合特征可以计算出印章图像的几何中心^[9]。另一种可选的方法则是利用边界链码跟踪外边缘，根据链码的统计平均值来计算几何中心。还可以通过旋转坐标系的方法多次计算几何中心，取中间的平均值，并计算出旋转角度。然后跟据模板图像的几何特征值和印章图像的边缘链码和几何中心计算出外边缘的匹配度，面积差值，周长差值等

根据印章的形状和顶点坐标值还可以计算出除圆形印章外的旋转角度，即旋转印章图像至与原始图像相一致的偏转角度。对于椭圆形等有上下对称特征的顶点坐标的图像还要统计图像的Y方向投影图，并与原始印章图像进行比较，若不一致则进行上下翻转。对于圆形印章图像，可以采用转动惯量最小化或者旋转匹配的方法来确定倾斜角度^[8]。

1.3 印章特征提取遇到的困难

在印章特征提取过程中，我们会遇到很多困难。一是印章不清晰。这主要是因为印泥过多或用力过重而发生字符与边框粘连，以及印泥过少或用力过轻而产生外边框轮廓断裂，还可能因为扫描到的印章图像背景过于复杂造成的。二是印章本身排版的复杂性。印章不仅有各种形状，其内容还包括图像、数字、汉字等多种成分，其中的文字区域与普通文档的排列不同，不仅有横向排列，还有弧形的。三是盖印时可能发生倾斜甚至翻转的情况，以至于扫描到的印章图像还需要倾斜校正等处理。上述的种种情况增加了印章特征提取得困难性。当前有很多学者也研究了在有上述这些干扰的情况下的有字符图像^[10]的字符区域定位提取方法，提出了最小面积法^[11]以及基于自组织映射神经网络的粘连字符分割^[12]等粘连字符分割方法等字符分割^[12-13]方法和定位^[14-18]方法。我们也将研究这些方法来解解决印章特征提取^[19-20]的问题。

1.4 本文的主要内容和结构

本文主要研究印章特征提取的一系列方法，包括印章图像的二值化算法、几何参数提取算法、字符区域提取算法等内容。本文将沿着图 1-3 所示的路线展开研究工作。在结构上本文共分为六章，具体安排如下：

第一章对印章自动识别系统作了一个概要的介绍，并且阐述了本文的研究背景和现实意义；介绍了印章识别系统的体系架构以及开发的详细流程；研究了当前印章识别中的几个主要模块；最后列出本文主要的研究内容和研究路线。

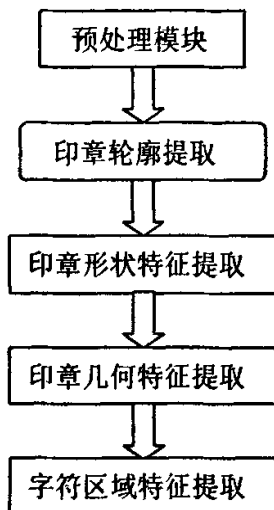


图 1-3 本文的研究路线

第二章针对印章图像的预处理模块的三个步骤进行了介绍，这三个部分依次是：提取印章颜色特征，二值化和去噪处理。印章的颜色为红色或者蓝色，我们针对这个特点采用了一种比较适合提取印章颜色特征的方法。图像二值化部分对当前比较常用的全局阈值法和局部动态阈值法进行了研究和比较，最后使用了一种基于 LOG 算子的局部动态阈值的算法。接着针对印章噪声点多的特征采用了比较适合的去噪方法。在这一章的最后对这些算法的效果进行了演示。

第三章针对规则印章形状的特点，通过提取印章的轮廓以及轮廓的端点，从得到的轮廓端点的个数和几何位置来确定印章的形状，并对实验的结果进行了统计。

第四章对方形、圆形、椭圆形印章的几何特点，提取它们的重要几何特征，对方形和椭圆形印章进行了倾斜校正。最后对印章比较重要的几何参数进行了精确性计算。

第五章针对印章字符的排版进行了介绍，并针对不同的字符排列特点采用了相适应的方法对字符区域特征进行了提取和定位。

第六章对全文进行了总结，指出了论文中的创新点和不足之处，并指出了进

一步的研究方向。

第2章 图像的预处理

2.1 引言

印章特征提取首先必须做的是将扫描到的印章图像从有很多噪声或干扰的不理想的图像模型转换为适合于识别的理想模型。这个阶段就是图像的预处理阶段，预处理阶段在提取印章特征过程中占有重要的地位，预处理做的好坏直接关系到我们后来的工作能否取得好的成果。印章图像预处理阶段包括确定印章颜色、二值化以及平滑去噪等过程。

印章区域图像的二值化^[21]是印章特征提取的关键。它对于提取印章几何特征、文字特征等等有着最直接的联系。在从灰度图像向二值图像的转化过程中，如何尽量使损失的信息量减少到最小，是人们长期以来所追求的目标。所以研究二值化算法的根本目的在于提取信息，将有用的部分作为前景显示出来与背景进行有效的区分。本章通过分析、比较多种二值化算法，采用了一种有效的算法来满足印章识别系统的需求。

印章的平滑去噪也很重要，二值化之后的图像可能仍存在许多噪声，影响之后的对形状、几何参数和字符区域特征的提取。本章分析了当前几种常用的去噪方法，采取了适合于印章图像的去噪方法。

2.2 印章颜色特征提取

2.2.1 真彩色图像

当前很多图像都是利用扫描仪或者是通过视频截取得到的，所以预处理前都是真彩色图像。真彩色图像也叫RGB图像。RGB图像中每一个像素的元素都是由R、G、B三个分量合成的。R、G、B分别表示红色、绿色和蓝色，它们也叫做RGB三原色，我们可以用它们合成任何一种颜色。真彩色图像中R、G、B的范围都是0~255。如黑色是(0, 0, 0)，白色为(255,255,255)，红色为(255,0,0)，蓝色为(0,0,255)等。

2.2.2 提取印章颜色特征算法

提取到印章图像后，我们要做的第一件事是提取它的颜色特征。我们提取到的印章图像为真彩色图像。众所周知，印泥的颜色通常为红色和蓝色。RGB空间中R代表红色，B则代表蓝色。背景色一般是白色，同时可能还存在一些黑色的噪

声点，然而黑色和白色的RGB值中 $R=B$ 。故如果印章是红色的，则图像中R值的累加和则远远大于B；否则R的累加和远小于B。

根据上述事实，本章采取下列方法来提取印章的图像。设立两的累加器 r, b ，类型都为 $double$ 。从上到下，从左到右扫描每一个像素点，对每一个像素点，我们将该像素点的R值累加到累加器 r 中，B值累加到累加器 b 中。为了防止累加器溢出，在累加时，分别对R和B值进行归一化处理。

扫描完毕之后，比较 r 和 b 的值。如果 r/b 大于1，并且 $r-b$ 大于某一个阈值 $threshold(threshold$ 至少应为100)，则印章颜色为红色；否则为蓝色。

2.3 图像的二值化

由于盖章时的力度问题可能会造成印章的不清晰，或者因为印泥的不均匀而产生很多噪声点，因此通过扫描仪扫描到的图像可能是亮度不同的多值图像，把多值图像转化为只具有黑白两种颜色的图像的过程就叫做图像的二值化^[22]。

印章图像二值化结果的好坏将直接关系到后面对印章几何特征和字符区域的特征提取的准确与否。二值化的一般方法分为全局阈值的二值化算法和局部自适应的二值化算法。

全局阈值的二值化算法是指在整幅图像中使用一个统一的阈值对灰度图像进行二值化。当图像具有单一的背景，并且图像灰度直方图明显呈现双峰分布时，采用全局阈值二值化算法一般可得到比较满意的结果^[23]。

在实际情况下，图像背景的灰度值可能并不是常数，而且前景和背景的对比度在图像中也不是固定的。如果图像亮度不均匀、有突发噪声，或者背景灰度变化较大时，必须根据像素的坐标位置关系自动确定不同阈值。由像素的灰度值和该像素点周围的像素局部特性来确定像素的阈值，从而进行二值化的方法称为局部阈值算法。

下面是二值化过程中常用到的两个概念。

1) 阈值(threshold)

设图像中某一像素 (x,y) 的灰度级别为 $f(x,y)$ ， t 为阈值，则 $f(x,y)$ 和 t 满足式(2.1)所示的关系。

$$f(x,y) = \begin{cases} 255, & f(x,y) > t \\ 0, & f(x,y) \leq t \end{cases} \quad (2.1)$$

该式表示，当点 (x,y) 的值大于 t 时，该点为背景点，否则该点为前景点。当然根据具体需求的不同，当灰度值大于 t 时像素也可以是前景点，小于 t 的为背

景点。这可以根据我们的需求而定。

2) 直方图 (Histogram)

对应于每个灰度值, 求出在图像中具有该灰度值的像素数的图形叫做灰度直方图, 或简称直方图。

2.3.1 常用的二值化算法

常用的二值化算法很多, 而我们只研究其中几种算法。我们将对下面几种算法进行简要分析和研究。

- 1) 双峰直方图阈值分割法和平均灰度法^[7]
- 2) Ostu 方法^[24]
- 3) 边缘算子法^[7]
- 4) Chow 和 Kaneko 的方法^[7]
- 5) Parker 的方法^[25]

其中 1), 2)和 3)是全局阈值的二值化算法, 而 4)和 5)则是局部自适应的二值化算法。

2.3.1.1 双峰直方图阈值分割法和平均灰度法

通过投影得到图像的灰度直方图, 如果直方图的前景和背景分别对应的投影形成两座明显的山峰, 那么这两个山峰中的最低谷处就可以作为二值化分割的阈值。这种方法称为双峰直方图阈值分割法。

对灰度变化不明显有两个山峰的, 也可以采用平均灰度法。图像的平均灰度可以通过先求整个图像各像素灰度的和, 再除以像素数而求得。以这个灰度值作为阈值, 就可以将灰度图像转变成二值图。

2.3.1.2 Ostu 方法

Ostu方法和双峰直方图阈值分割法以及平均灰度法都是全局阈值的二值化算法。Ostu方法实际是根据最小二乘法的思路演变而来的。如图像中的所有像素的灰度级集合为 $\{p_0, p_1, \dots, p_k, \dots, p_n\}$, 其中当 $i < j$ 时, $p_i < p_j$; 假定 p 为前景点和背景点的分割点, 即如果 $p_i > p$, 则灰度级为 p_i 的点为前景点, 否则为背景点。

设图像中前景点所占的比例为 u_0 , 前景点的平均灰度为 w_0 ; 背景点所占的比例为 u_1 , 背景点的平均灰度为 w_1 , 则平均灰度 w 的值可以由式子(2.2)来计算。

$$w = u_0 \times w_0 + u_1 \times w_1 \quad (2.2)$$

根据最小二乘法的启示, 遍历灰度 p , 当 p 使得下式中 g_0 为最大时, 即为最满意的分割阈值。

$$g_0 = u_0 \times (w_0 - w)^2 + u_1 \times (w_1 - w)^2 \quad (2.3)$$

式子(2.3)实际上就是类间方差值。而前景取值 w_0 , 概率为 u_0 , 背景点所占的比例为 u_1 , 前景点的平均灰度为 w_1 , 而总均值为 w , 根据方差的定义即得该式。因方差是灰度分布均匀性的一种度量。方差值越大, 说明构成图像的两部分差别越大。我们不妨按照极限情况去考虑, 当全部的点都是前景点或者都为背景点时, g_0 为最小值0。实现ostu算法时候, 直接应用 g_0 的表达式计算量过大, 因此实现过程用了与(2.3)等价的公式(2.4)。这种方法实际上是以中最大类方差算法。算法的流程图如图2-1所示。

$$g_0 = u_0 \times u_1 \times (w_0 - w_1)^2 \quad (2.4)$$

2.3.1.3 边缘算子法

采用Laplace算子、Robert算子、Sober算子等对像素点进行灰度级增强或减弱的变换。对于灰度均匀分布的区域内的像素点, 利用这些算子可以对其进行灰度减弱; 而对于在边缘附近的像素点, 这些算子可以对其进行灰度增强。

2.3.1.4 Chow 和 Kaneko 的方法

Chow和Kaneko于1972年提出的通过灰度分布的局部信息来计算整幅图像的自适应阈值面的二值化方法是一种局部阈值算法。这种方法是将整幅图像划分成为许多互不重叠的并且面积相等的小块, 组成规则的网格。对每一个网格进行投影, 并且进行双峰测试。从具有双峰的直方图中取得对应网格中的局部阈值, 并对这些阈值进行插值计算, 最后得到整个图像的阈值面。

这种算法对于只能处理图像背景比较单一而且图像灰度直方图明显呈现双峰分布的全局阈值二值化方法来说, 确实有了很大的改进。然而, 实现起来却有一定的难度。首当其冲的就是图像上网格划分的大小问题: 如果网格面积很小, 其产生的直方图的起伏就不明显, 得到的局部阈值可能是无效的。另外, 网格图像如果落在全背景或者全前景区, 则仍然会呈现局部灰度图像的双峰分布。这样取得的局部阈值将对结果造成非常大的影响, 可能会导致严重的错误。

2.3.1.5 Parker 的方法

这种方法是先找出边界点, 然后根据边界点来确定其他点的阈值。具体的算

法如下。

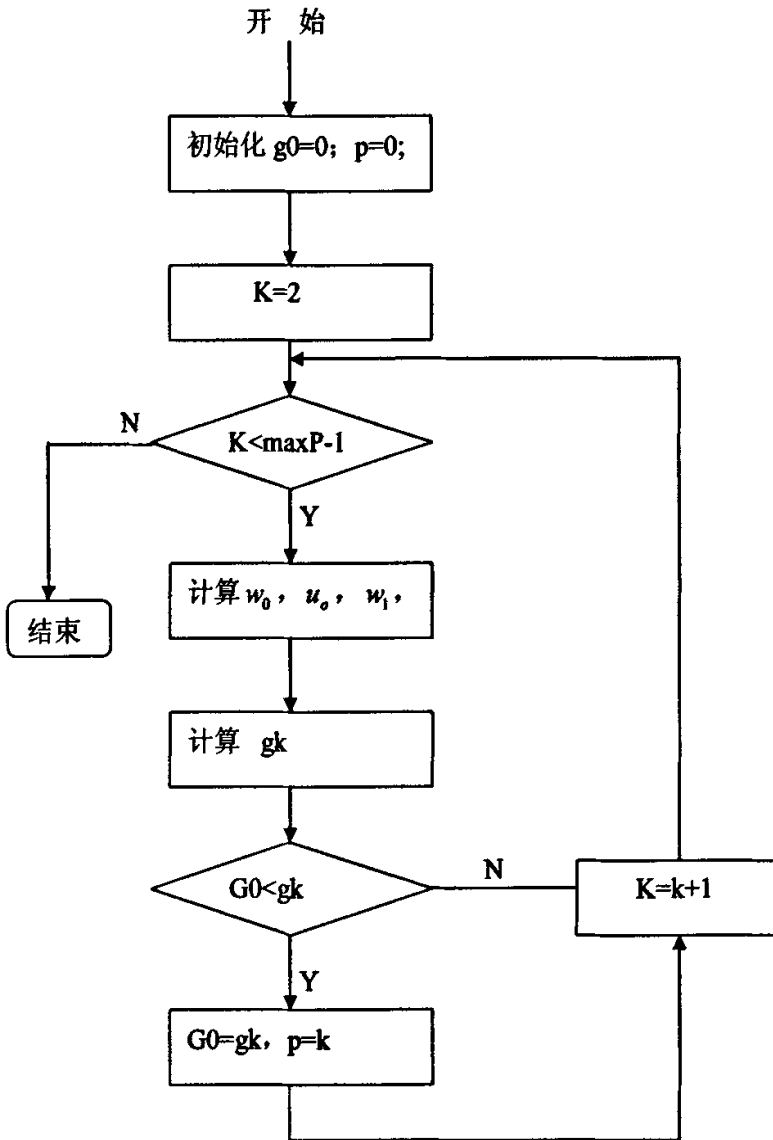


图 2-1 Ostu 算法流程图

1) 对每个像素(x,y)，计算出它和其8个邻域点的最小灰度差值D(x,y)，如式(2.5)所示，这样将找出一个最亮的负的梯度值。

$$D(x,y) = \min_{i=1..8} (G(x,y) - G(x_i,y_i)) \quad (2.5)$$

2) 将D图像分解成 $N*N$ 大小的子区域,计算出每一个子区域的样本均值和标准差。其中参数 N 取16。

3) 首先对所求得的均值和标准差进行平滑处理。然后对得到的每个区域的均值和标准差进行插值,计算出各点的梯度均值 M 和标准差 S 。对像素点 (x,y) ,如果 $M(x,y)>-1$ 或 $S(x,y)<1$,则像素 (x,y) 为未标记类;如果满足式(2.6)的要求,则像素 (x,y) 为前景点,否则 (x,y) 仍归为未标记类。此处的常数 k 常取-1值。

$$D(x,y) < M(x,y) + S(x,y) \quad (2.6)$$

4) 对尚未标记的像素 (x,y) ,找到它的八邻域中已经归于前景类的像素的灰度级最小值。如果 (x,y) 的灰度级小于这个值,则将它归于前景类。

5) 重复步骤4,直至找不到更多的前景点为止。

2.3.2 本文采用的方法

我们现在是对已经从纸质或其它质料上提取出来的印章图像进行处理,即已经对原始扫描图像进行了去除背景的处理。要处理的样章如图 2-2 所示。从图 2-2 上可以看到,图像的轮廓边缘往往会出现不清晰或者局部断裂的现象。因此我们需要在二值化过程中要注重对图像边缘进行增强。并且需要对字符的细节信息进行刻画。Laplace 算子能够对图像的边缘进行灰度增强,然而 Laplace 算子对噪声比较敏感。所以用了 Laplace 算子对图像边缘进行增强的同时,可以用滤波器对噪声进行平滑。Gauss 滤波是常用的一种图像平滑去噪方法。将两者结合起来就可以取得我们想要的效果。因此,我们在这里利用了 Laplacian-Gauss 算子,即 LOG 算子来对图像进行灰度处理和二值化。



图 2-2 处理样章

2.3.2.1 LOG 算子介绍

虽然梯度算子和 Laplacian 算子能够对图像的边缘进行增强处理,但它们都对噪声比较敏感。为了利用这些算子进行边缘提取,必须消除噪声对边缘提取的影响。一种做法是在运用这两种算子作边缘提取前,先用邻域平均法作平滑处理;另一种做法则是先用高斯二维低通滤波器对图像进行滤波,然后再对图像作 Laplacian 边缘提取。后者被称之为 Laplacian-Gauss 算子法,即 LOG 算子法。二维高斯低通滤波器的表达式如(2.7)所示。

$$G(x, y) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (2.7)$$

其中 (x, y) 是图像中像素的坐标,参数 σ 是概率分布函数的标准方差。通常式(2.7)的前面会加上一个规范化的因子,如公式(2.8)所示。

$$G(x, y) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (2.8)$$

其中 σ 是高斯滤波器的唯一参数,它同滤波器作用邻域的大小成正比。像素距离高斯滤波器算子中心越远,对滤波器的影响越小;如果像素距滤波器的中心的距离达到 3σ ,这时像素对滤波器的影响可以忽略不计。

如果一幅图像先经过高斯滤波,再经过 Laplacian 算子变换,这个操作可以缩写为 LOG (Laplacian of Gaussian)。Laplacian 算子就是二阶导数 $\nabla^2 f(m, n)$,该算子与方向无关。LOG 算子可以写成如(2.9)的卷积形式。

$$\nabla^2[G(x, y, \sigma) * f(x, y)] \quad (2.9)$$

由于 Laplacian 算子是线性的,上面这个卷积式子的微分和卷积运算可以相互交换,因此上式也可以写成如(2.10)的形式。

$$\nabla^2[G(x, y, \sigma)] * f(x, y) \quad (2.10)$$

由于高斯滤波器的二阶导数 $\nabla^2 G$ 同具体的图像无关,因此可以预先计算它,以降低实际应用的复杂度。为了简化起见,我们用 r^2 表示 $x^2 + y^2$,这里的 r 就是像素点距高斯滤波器中心的距离,这样的替代是很合理的,因为高斯函数是关于原点成中心对称的,并且这样的替换使得二维的空间变为一维空间,易于对其进行微分运算。

$$G(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right) \quad (2.11)$$

对公式(2.11)进行一阶微分,得到的结果如式(2.12)所示,其二阶微分得到的结果如式(2.13)所示。

$$G'(r) = -\frac{1}{\sigma^2} r \exp\left(-\frac{r^2}{2\sigma^2}\right) \quad (2.12)$$

$$G''(r) = \frac{1}{\sigma^2} \left(\frac{r^2}{\sigma^2} - 1\right) \exp\left(-\frac{r^2}{2\sigma^2}\right) \quad (2.13)$$

我们再返回到(x, y)坐标, 并且引进一个规范化的乘积因子 c, 于是就得到了 LOG 算子的卷积模板, 如式(2.14)所示。

$$h(x, y) = c \left(\frac{x^2 + y^2 - \sigma^2}{\sigma^4}\right) \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (2.14)$$

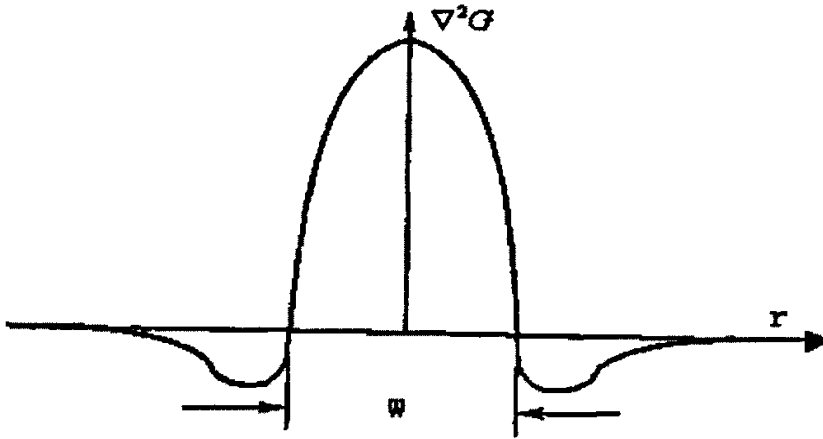


图 2-3 $\nabla^2 G$ 的截面图, 其中 W 为主瓣宽度

乘积因子 c 规范模板元素之和为 0。LOG 算子图的形状如墨西哥草帽, 见图 2-3。在实际应用中 $\nabla^2 G$ 可选取如(2.15)的 5×5 模板。也可选取比较大的模板, 较常见的有 17×17 的模板。所选取的模板窗口尺寸与 $\nabla^2 G$ 的图形的主瓣宽度有关, 如图 2-3 所示, 窗口模板内的系数和为零, 边缘与卷积计算后的零交叉点对应。

$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix} \quad (2.15)$$

2.3.2.2 基于 LOG 算子的局部动态阈值算法

该算法可描述如下：初始化 LOG 算子模板。这里我们采取的是 11×11 的模板。从上到下，从左到右扫描每一个像素。对每一个像素进行 LOG 算子变换。

```

1. 初始化向量 tempImg(M,N)所有元素为 0;
2. 初始化  $11 \times 11$  的 LOG 算子模版 LOG(11,11);BASE=5;
3. 初始方向 dir=4;
4. FOR i=BASE;i<M-BASE ;i++DO
5.     FOR j=BASE;j<N-BASE ;j++ DO
6.         FOR di=-BASE;di<=BASE;di++ DO
7.             FOR dj=-BASE;dj<=BASE;dj++ DO
8.                 temp(I,j)+=Img(I+di,j+dj)*LOG(di,dj);
9.             END FOR
10.        END FOR
11.    END FOR
12. END FOR
13. FOR i=0;i<M ;i++DO
14.    FOR j=0;j<N-BASE ;j++ DO
15.        IF temp(i,j)<255 DO
16.            Img(I,j)=0;
17.        ELSE
18.            Img(I,j)=255;
19.        END IF
20.    END FOR
21. END FOR

```

图 2-4 基于 LOG 算子的局部动态阈值算法

图 2-4 中第 1 行中的 M , N 分别代表图像的长度和高度。第 9 行的向量 *Img* 指的是输入的印章图像。图 2-5 中的 (a) 双峰直方图全局阈值分割法二值化后的结果, (b) 是 Laplacian 算子法二值化后的结果。和图 2-6 中基于 LOG 算子的局部动态阈值算法的二值化效果对比。我们不难发现基于 LOG 算子的局部动态阈值算法在进行二值化处理之前首先会对图像的边缘信息进行增强, 而当前像素的最终

灰度的信息还取决于周围邻域内的像素的信息，从而顾及了图像的局部信息。对于一些字符的细节信息也能够很好地体现。达到了很好的效果。

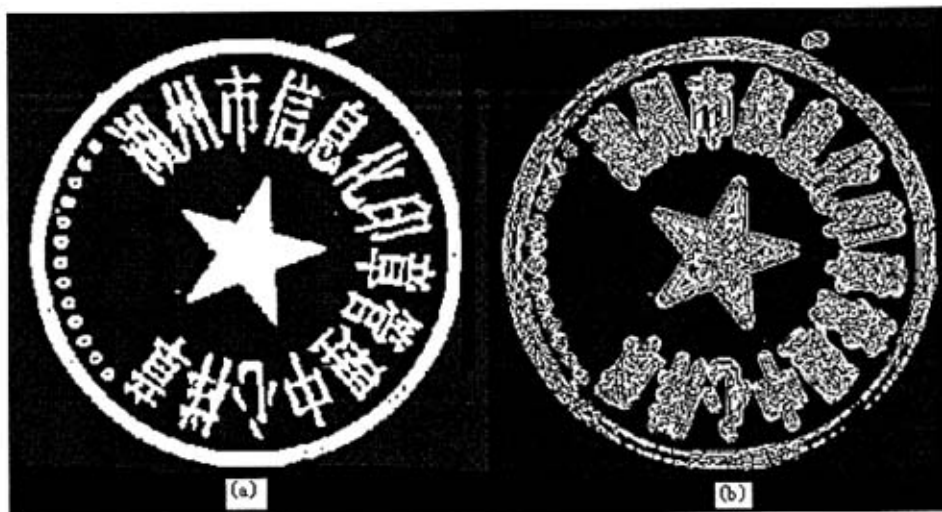


图 2-5 双峰直方图阈值分割法(a)和 Laplacian 算子法(b)的二值化效果



图 2-6 基于 LOG 算子的局部动态阈值算法的二值化效果

当然，虽然本算法在效果上比全局阈值算法以及其他的局部动态阈值算法要好很多，但是它也会随之带来两个不利影响：

1) LOG 算法运算需要的时间长，这对于印章的识别显然是非常不利的。而金融部门或者公安部门对印章的鉴别对于实时化的要求会越来越高。然而二值化

的好坏却是直接关系到印章识别后面的所有工作的关键，因此利用 LOG 算子进行二值化还是可取的。

2) LOG 算子过多的强调、突出细节信息会引入很多噪声点，这样就要求我们必须对二值化后的图像进行去噪处理。这样我们必须找到一种有效去除噪声点的方法来达到我们的目的。

2.4 图像的去噪方法

消除噪声都是采用低通滤波的方法。低通滤波在消除图像噪声的同时，也会消除图像部分有用的高频信息。因此，各种图像去噪方法，其实就是在去噪和保留有用高频信息之间进行权衡。

2.4.1 常用的去噪方法

1) 基于中值滤波的方法

中值滤波的方法是常用的图像去噪方法。它的基本原理是按照某种方法取当前像素周围八邻域中的一个点的灰度值作为当前点的灰度值。这种算法简单而又易于计算。

2) 基于神经网络的方法

神经网络^[17]是由大量的神经元广泛互连而成的网络。有几位学者提出了一种能够根据噪声环境自适应的调节多层神经网络的参数的基于中值滤波器和多层神经网络的方法，这种方法对高斯噪声和脉冲噪声有良好的去噪效果。

3) 基于维纳滤波的方法

维纳滤波器是一种线性滤波器，主要是用线性代数的方法对图像进行处理。它具有约束恢复的能力，只须有噪声均值和方差便可计算出最优结果。

2.4.2 本文用的去噪方法

从前面的二值化过程可知，利用 LOG 算子进行图像的二值化处理由于过度强调局部细节的保存，因而产生了很多噪声点。如何消去这些噪声点是我们要考虑的问题。从常用的图像处理方法中分析得知，基于中值滤波的去噪方法可以很好地解决图像中的噪声点问题。噪声点实际上就是图像中的离散前景点。它的特征是其八个邻域都是背景点。

我们所用的去噪方法描述如下：从上到下，从左到右扫描图像，如果当前点是前景点，但其八个邻域都是背景点，说明当前为噪声点，应该消去。如果印章中噪声过多还可以对印章图像进行多次迭代去噪处理。算法的伪代码如图2-7所

示。

```

1. int a1,a2,a3,a4,a5,a6,a7,a8;
2. FOR i=0;i<N;i++ DO
3.     FOR j=0;j<M;j++ DO
4.         IF temp(j,i)=0 DO
5.             a7=temp(j,i-1) &temp(j,i+1)& (temp(j-1,i)|temp(j+1,i));
6.             a8= temp(j+1,i) &temp(j-1,i)& (temp(j,i-1)|temp(j,i+1));
7.             temp(j,i) = a7|a8;
8.         ELSE
9.             a1=0x00000001^(temp(j,i-1)|temp(j,i+1));
10.            a2=0x00000001^(temp(j-1,i-1)|temp(j-1,i)|temp(j-1,i+1));
11.            a3=0x00000001^(temp(j+1,i-1)|temp(j+1,i-1)|temp(j+1,i+1));
12.            a4=0x00000001^(temp(j+1,i)|temp(j-1,i));
13.            a5=0x00000001^(temp(j-1,i+1)|temp(j,i+1)|temp(j+1,i+1));
14.            a6=0x00000001^(temp(j-1,i-1)|temp(j,i-1)|temp(j+1,i-1));
15.            temp(j,i) = 0x00000001 ^ ((a1&(a2|a3)) | (a4&(a5|a6)));
16.        END IF
17.    END FOR
18. END FOR

```

图 2-7 论文中使用的去噪算法伪代码

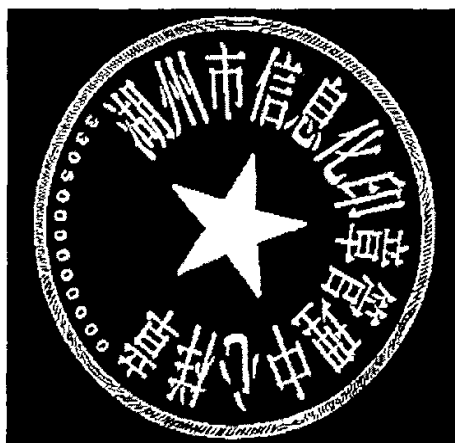


图 2-8 去噪后的效果图

这种方法实现简单而且快速，去噪效果也很好。去除噪声后的效果图如图2-8所示。对比图2-6我们可以看出，二值化后产生的噪声点已经基本上被清除。

预处理之后的印章图像清晰，粘连情况也大为减少，字符边缘的细节也得到了很好的体现。为我们对形状特征，几何特征以及字符区域特征的提取提供了很好的保证。

2.5 本章总结

本章研究印章识别系统中的预处理算法。首先提取印章的颜色信息。针对印章颜色只有红或者蓝两种颜色采用了与之对应的方法确定印章的颜色。

接着对图像进行二值化处理。并对目前很多成熟的二值化算法——主要包括两大类：全局阈值法和局部动态阈值法——进行研究、分析出发，对其中一些常用的二值化算法的原理进行了剖析。本文在研究已有二值化算法的基础上采取了一种基于 LOG 算子的局部动态阈值的算法来进行图像的二值化。基于 LOG 算子的局部动态阈值算法通过边缘增强可以凸显出字符的细节信息，有利于印章中字符的识别。但是也由于过分注重突出局部细节从而增加了图像的噪声点，给我们的除噪工作增加了工作量。

最后对当前去噪的主要算法进行了阐述和研究，采取了一种适合于印章图像的去噪处理方法。并对最终取得的预处理效果进行了演示。

第3章 印章形状特征提取

3.1 引言

印章主要有方形、圆形、三角形、椭圆、菱形等典型的凸多边形几何形状。在印章的特征提取过程中,对其形状特征的提取是其中相当重要的一步。印章形状特征提取的正确性关系到对印章的校正、分割和识别的准确率^[27]。

3.2 印章图像轮廓线的提取

轮廓提取的方法有很多种,我们这里主要研究基于扫描线的轮廓提取方法和链码跟踪提取轮廓方法。这里的轮廓提取指的是对印章外轮廓线的提取,我们将选用其中一种方法进行轮廓提取。

3.2.1 基于扫描线的印章轮廓提取

该方法描述如下:采用三条扫描线分别从水平和竖直方向对印章图像进行扫描,对扫描到的轮廓点进行剔出歧点操作,并且将两个方向扫描到的结果进行合并。对得到的轮廓点进行中值滤波去除边界断点,从而得到印章的完整边界。其示意图如图 3-1 所示。基于扫描线^[6]的印章轮廓提取方法能够得到比较平滑的边界。

3.2.2 边界链码跟踪边缘轮廓提取方法

相邻像素之间的两种位置关系分别是 8-邻域和 4-邻域。这两种位置关系可以简单描述如下。

1) 像素的 4-邻域(4-Neighbor)。对像素 (i, j) , 该像素自身和图像中的与其相邻的上,下,左,右 4 个像素所构成的集合,称作像素 (i, j) 的 4-邻域,有时也称 4-近邻。

2) 像素的 8-邻域(8-Neighbor)。对像素 (i, j) , 该像素自身和图像中的与其相邻的上,下,左,右 4 个像素以及对角线的 4 个像素所构成的集合,称作像素 (i, j) 的 8-邻域,有时也称 8-近邻。

像素 (i, j) 的 4-邻域有时也称为像素 (i, j) 的直接邻域,其符号表示为 d -近邻。一个像素的 8-邻域除去它的 d -邻域,余下的对角线上的 4 个像素,则称为像素 (i, j)

的非直接邻域，取符号表示为 i -近邻。

一个简单二值图像闭合边界的轮廓跟踪算法很简单：首先按从上到下，从左到右的顺序搜索，找到的第一个前景点一定是最左上方的轮廓点，记为初始点 A 。它的右，右下，下，左下四个邻域点中至少有一个是轮廓点，将这个点记为 B ，将 B 点保存于轮廓向量中。从 B 点开始跟踪寻找，按右，右下，下，左下，左，左上，上，右上的顺时针顺序找其 8 邻域中第一个遇到的轮廓点 C 。如果 C 就是 A 点，则表明已经转了一圈，程序结束；否则保存 C 点于轮廓向量中，并从 C 点继续跟踪寻找，直到找到 A 为止。至于判断是不是轮廓点很容易：如果一个像素的直接邻域都是背景点则该点不是轮廓点，否则是轮廓点。这种算法对尺寸大于 1 个像素的所有区域都是有效的。

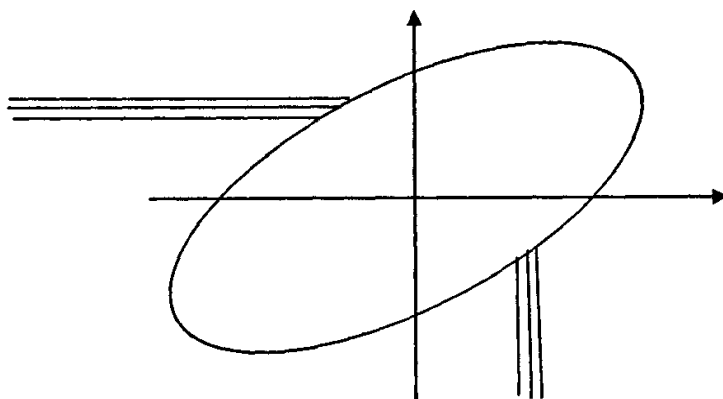


图 3-1 基于扫描线的轮廓提取算法示意图

为了得到平滑的图像轮廓，可以在检测和跟踪图像轮廓后，利用轮廓点的方向信息来平滑轮廓。显然，图像轮廓噪声越大，图像轮廓点变化越剧烈，图像轮廓相邻点的方向变化数也越大。根据这一特点，设置一个轮廓点方向变化数阈值。把方向变化数大于这一阈值的图像轮廓点滤除，由此可得到平滑的图像轮廓。轮廓跟踪常常跟轮廓平滑一起使用。

3.2.3 本章所使用的方法

由于我们研究的印章图像都存在一个封闭的外轮廓，并且该轮廓都是凸多边形的，因此我们采用 8-邻域的链码跟踪算法来提取印章轮廓。我们采用的算法如图 3-2 所示。

图 3-2 的第 1 行, 初始像素 m_start_point , 可以用很多方法找到。通常可以通过由上至下, 由左至右扫描得到。跟踪过程可以看成是沿属于该集合的像素向其邻近的区域走。这里我们选取的是 8-邻域的方向。因为圆形和椭圆形印章的边界并非直线分布, 就算是方形的印章, 可能由于盖印时发生倾斜, 因此 4-邻域的链码跟踪算法不适合印章轮廓的提取。第 2 行的搜索方向 $testdir$ 的范围是 0-7。对起始点 m_start_point , 我们从右上邻域点即方向 dir 为 4 开始, 沿顺时针方向扫描其 8-邻域点直到扫描到一个当前点为止, 并将该点作为下一个当前点 $current$ 。对每一个当前点 $current$ 我们总是选取从其上一个当前点按顺时针扫描到的第一个前景点作为下一个当前点。并且整体的方向是顺时针的。在搜索过程中, 一般来说当前点 $current$ 不是起始点, 因此若当前像素和初始像素相同时, 跟踪就结束了。

```

1. 扫描得  $m\_start\_point$ ;
2. 设置  $current=m\_start\_point$ , 搜索方向  $testdir$ 
3. 初始方向  $dir=4$ ;
4. DO
5. 初始化搜索方向  $testdir=(dir-3)\&0x07$ ;
6. DO
7. IF
8.      $current$  的( $testdir$ )方向的邻域点 A 在轮廓上 THEN
9.     置当前点  $current=A$ ;
10.    Break;
11. ELSE
12.    改变搜索方向  $Testdir=(testdir+1)\&0x07$ ;
13. END IF
14. WHILE(1);
15. Dir 设置为  $testdir$ ;
16. Current 加入轮廓向量  $bound$ ;
17. WHILE( $current\neq m\_start\_point$ );

```

图 3-2 链码跟踪提取轮廓算法

本章应用的轮廓提取算法充分利用了印章轮廓的封闭性特征。8-邻域链码跟

踪算法快速而且高效。尽管印章的边缘可能仍存在少量断裂或者噪声，但经过多次试验发现，利用该算法总能得到完整的轮廓，有利于下述印章形状的确定。利用8-邻域链码跟踪算法得到的印章轮廓如图3-3中的红线所示。从图3-3可以看出，尽管印章左下角的外边缘有一个断裂，该算法仍然可以通过跟踪断裂处的前景点形成一个完整的轮廓。

3.3 印章形状确定算法

得到印章的轮廓之后，就可以根据取得的轮廓向量来确定印章的形状。由于印章的形状固定，只有方形、圆形、椭圆形、菱形、三角等几种图形。经过观察可以得出，形状不同，其顶点的个数也各不相同，方形和菱形有四个顶点，圆形与椭圆形印章顶点大于四，而三角形印章顶点则只有三个。可以通过对印章的封闭轮廓进行一系列处理，从而得到轮廓的顶点个数。利用顶点的个数信息来判定印章的形状。得到轮廓定点的方法是去除轮廓点之中满足下列条件的点：

如果轮廓向量中的相邻轮廓点 $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$ ，如果向量 \overline{BA} 和向量 \overline{CB} 之间的夹角小于某一个阈值 $threshold$ ，则将 B 点从轮廓向量中去除。为了求这个夹角，我们将利用离散余弦公式(3.1)。

$$\cos\theta = \frac{a^2 + b^2 - c^2}{2ab} \quad (3.1)$$

直接利用离散余弦公式，会有很多的计算量。为了尽量提高效率。我们首先对离散余弦公式做了如下的优化处理。

向量 \overline{BA} ($x_2 - x_1, y_2 - y_1$) 设为 $(M1, N1)$ ，向量 \overline{CB} ($x_3 - x_2, y_3 - y_2$) 设为 $(M2, N2)$ ，其夹角为 θ ，则利用余弦公式得到如式(3.2)的等价计算公式。利用该式，将减少很多计算量。

$$\cos\theta = \frac{(M1 \times M2 + N1 \times N2)}{|\overline{BA}| \times |\overline{CB}|} \quad (3.2)$$

对轮廓向量作上述多次去除中间点的迭代，直到剩下的所有点中任意相邻的三个点 A , B , C ，都满足向量 \overline{BA} 和向量 \overline{CB} 之间的夹角都大于某一个阈值 $threshold$ 为止。这些点就是我们要求的印章图像顶点。如果顶点小于3，说明是失败的，可能并不存在印章。如果顶点等于4，则说明是菱形或者方形。还需要进一步确定其形状。如果顶点等于3，则说明是三角形印章。如果顶点大于4，则说明是椭圆形或圆形。还要根据扁率来确定其真实形状。

该算法简单而且高效，通过对余弦公式的先验改进减少了计算量。通过实验，

效果非常好。能够准确地确定印章的形状。

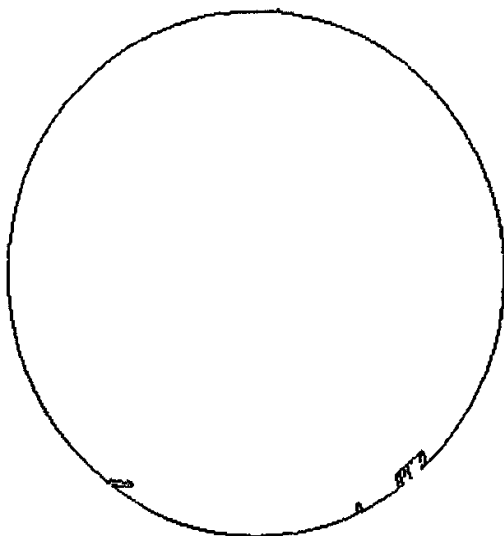


图 3-3 外边缘有断裂的印章的轮廓提取结果

3.4 本章小结

本章首先提取了印章的轮廓，根据轮廓端点的个数确定了印章的形状。经过实验得到结果如表 3-1 所示。从表 3-1 可以看出，本章所采用的方法提取印章形状的正确率达到 93%。因此该方法对提取印章形状特征是有效的。

表 3-1 印章形状特征提取结果

印章图像个数	正确个数	错误个数	正确率
30	28	2	93%

第4章 印章几何特征提取

4.1 引言

在上一章中我们提取了印章的外轮廓，并且通过提取到的印章外轮廓的端点个数获得了印章的形状。在本章中，我们将根据印章的轮廓和端点得到印章的几何参数。如方形印章的倾斜角度，圆形印章的圆心和半径，椭圆形印章的原心，长短轴，曲率，倾斜角度等等。

最后由于印章的不清晰，边缘断裂等问题而造成我们得到的几何参数有一定的误差——如圆的圆心提取产生误差——而这些误差对我们后面的提取印章的字符区域的影响可能是致命的。因此我们这一章也将就如何修正这些误差进行研究，并且针对每一种印章的几何特性采取相应的修正方法。

本章主要就印章中有代表性的方形、圆形和椭圆形这三种印章进行探讨和研究，并列出了解决方案。

4.2 初步提取几何参数特征

对顶点为四的印章。可知其为菱形或方形。这个可以根据四个顶点的位置关系初步确定是菱形还是方形。一般相邻两条边大致相等，可确定为菱形或正方形。至于相邻两条边相差较大，则一定是长方形印章。我们保存其四个顶点作为几何参数和形状作为几何参数，这些将在下一章对印章的倾斜校正中用到。

对顶点为三的印章，则必定是三角形的。保存其三个顶点为其几何特征。对多个顶点的印章，则是圆形与椭圆形。我们可以分析轮廓向量来初步计算其几何特征以便于后来的精确计算。

下面我们将主要介绍圆形与椭圆形印章形心的计算方法和其它主要参数的计算，如圆形印章的半径，椭圆形印章的长短轴比率等，并以此来判定印章的形状。

4.2.1 确定圆形和椭圆的几何中心

根据圆形和椭圆的轴对称和中心对称特性。可以知轮廓向量中横坐标值最小点和最大点的平均值就是几何中心横坐标。轮廓向量中纵坐标值的最小和最大值的平均值就是几何中心的纵坐标^[28]。计算几何中心的算法如图 4-1 所示。

```

1. 轮廓向量为 bound, 几何中心(cx,cy);
2. 初始化  $\min X = \max X = \text{bound}[0].x, \min Y = \max Y = \text{bound}[0].y$ ;
3. 计数器  $i=0$ ;
4. DO
5.  $i++$ ;
6.     IF  $\min X > \text{bound}[i].x$  THEN
7.          $\text{Minx} = \text{bound}[i].x$ ;
8.     END IF
9.     IF  $\max X < \text{bound}[i].x$  THEN
10.         $\text{MaxX} = \text{bound}[i].x$ ;
11.    END IF
12.    IF  $\min Y > \text{bound}[i].y$  THEN
13.         $\text{MinY} = \text{bound}[i].y$ ;
14.    END IF
15.    IF  $\max Y < \text{bound}[i].y$  THEN
16.         $\text{MaxY} = \text{bound}[i].y$ ;
17.    END IF
18. WHILE  $\text{bound}[i] \neq \text{end}$ ;
19.  $\text{cx} = (\min X + \max X) / 2$ ;
20.  $\text{cy} = (\min Y + \max Y) / 2$ ;

```

图 4-1 计算几何中心算法

图 4-1 中算法第 1 行中的 *bound* 是通过链码跟踪算法提取的外轮廓点的向量。第 19 和 20 行的 *cx* 和 *cy* 是几何中心的横坐标和纵坐标。通过该算法得到的圆形印章圆心如图 4-2 所示。

4.2.2 计算圆形和椭圆的几何参数

计算轮廓向量上的所有点到几何中心的平均距离, 最大距离和最小距离。如果是圆形印章, 则平均距离与轮廓上的点到半径的距离显然相差不大。否则必定有较大差距。如果根据此特征来确定该印章形状是圆还是椭圆。算法如下:

- 1) 设轮廓向量为 *bound*, 几何中心(*cx,cy*);
- 2) 初始化最短距离 minLen =图像的宽度; $\text{maxLen}=0$; 计数器 $i=0$; 所有轮廓

点到图像中心点距离的和 $total_r=0$; 存储半径的向量 r ;



图 4-2 找到错误的几何中心

3) 求轮廓线上的点到形心的最大和最小距离，算法流程如图 4-3 所示。

```

1. WHILE bound[i] <> end DO
2.   计算当前点到几何中心的距离 len;
3.   IF len < minLen THEN
4.     minLen = len;
5.   END IF
6.   IF len > maxlen THEN
7.     maxlen = len;
8.   END IF
9.   Total_r 设置为 total_r + len;
10. 将 len 存入半径向量 r 中;
11. I++;
12. END WHILE;

```

图 4-3 计算轮廓点到形心的距离最大和最小值算法

4) 判定形状, 计算几何参数。算法如图 4-4 所示。图 4-4 的第 1 行首先求得轮廓线上的点到几何中心的平均距离 avg_r 。第 2 行中的 $count$ 是轮廓线上的点到几何中心的距离与平均半径的差距大于阈值的计数器。图 4-4 中第 4 行中的 $threshold$ 是一个经验值, 我们取 0.1, 即如果是圆形印章, 则轮廓线上的点与圆心的距离与平均半径相差不超过十分之一。鉴于我们求得的几何中心可能会有一

```
1. 平均半径  $avg\_r=total\_r/$ 轮廓向量中点的个数;
2. 设置  $i=0;$ 计数器  $count=0;$ 
3. WHILE  $r[i]<end$  DO
4.   IF  $abs(r[i]-avg\_r)/avg\_r>threshold$  DO
5.     Count++;
6.   END IF
7.   I++;
8. END WHILE;
9. IF  $count/$ 轮廓线上点的个数 $<threshold$  DO
10.  标记该印章为圆形;
11.  标记半径  $R=avg\_r;$ 
12.  设置圆心  $axis(cx,cy);$ 
13. ELSE
14.  标记该印章为椭圆;
15.  标记该椭圆长半轴  $a=maxLen;$ 
16.  标记该椭圆短半轴  $b=minLen;$ 
17. END IF
```

图 4-4 计算圆形与椭圆形印章几何参数算法流程

定误差, 因此如果不满足上述要求的点的个数占据了所有轮廓点个数的十分之一以上, 则肯定其为椭圆形的, 否则则是圆形的。然后根据相应的形状特点来求得其几何参数。如圆形印章的半径, 椭圆形印章的长短轴, 曲率等等。

通过试验发现这个算法对于面积较小的圆形印章或椭圆形印章非常有效, 能够准确找出几何中心和其它参数。然而如果印章的面积超过一定值则会出现很大误差。很大程度上影响了下面几章要进行的特征提取工作。例如图 4-2 是使用上述算法确定圆形印章几何中心的演示结果。图中的印章图像是原图像按照 1 比 3

的比例缩小而成的，可以看出其几何中心产生了偏移。

4.3 方形印章几何特征的修正和倾斜校正

观察方形印章可以看出其四边都是直线。我们已经得到了四个顶点，由于边缘断裂或者残缺这样的干扰，因此将这四个顶点顺序用直线链接大部分情况下并不能得到一个准确的方形。为了便于倾斜校正，我们使用检测方形印章外轮廓四条边的方法来得到方形四个顶点的正确坐标，并且在最后对其进行倾斜校正。

4.3.1 Hough 变换

Hough 变换^[8]非常适合于在图像中查找直线。Hough 变换的原理非常简单：在图像中如果有一条直线，原点到该直线的距离为 d ，直线的方向角为 θ ，如图 4-5 所示。从图 4-5 可知，该直线上的任意一点 $A(x,y)$ 都满足方程(4.1)。

$$d = x \cos \theta + y \sin \theta \quad (4.1)$$

Hough 变换找到直线的方法就是依据上述事实形成的。算法如下：首先定义一个直线结构 LINE。

```

Typedef struct {
    int minX;//最小横坐标
    int maxX;//最大横坐标
    int minY;//最小纵坐标
    int maxY;//最大纵坐标
}LINE;

```

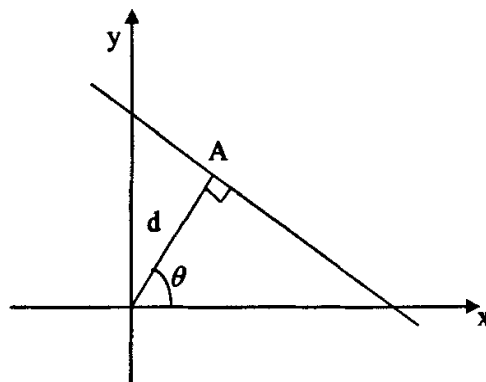


图 4-5 一条与原点距离为 s ，方向角为 θ 的直线

1) 初始化

初始化一个元素都为 0 的二维数组 $a[M][N]$ 做为计数器，第一维是角度，而第二维是距离。第二维的大小即点到直线的最大距离是图像对角线长度 M ， M 表示如式(4.2)所示。

$$M = \sqrt{nWidth^2 + nHeight^2} \quad (4.2)$$

其中 $nWidth$, $nHeight$ 分别为图像的宽度和高度。角度则从 0° 到 180° 之间变化。至于第一维的大小与所选的角度步长有关。如果选每次角度增加 1° ，则第一维就是 180，但一般我们选取的增量是 2° ，因此第一维的大小为 90。

初始化一个以结构 LINE 为元素类型的二维数组 $b[M][N]$ ； M , N 大小与数组 a 一样；初始化其中每一个元素 LINE，如式 (4.3) 所示。

$$\begin{aligned} \min X &= \min Y = \max(nWidth, nHeight) \\ \max X &= \max Y = -1 \end{aligned} \quad (4.3)$$

2) 从上到下，从左到右扫描图像，遇到前景点 $A(x,y)$ ，则顺序计算 A 点可能在的所有直线到原点的距离 d ，即按照 A 点所能在的直线的方向角 θ 从 0° 到 180° 之间，以 2° 的增量顺序计算直线到原点的距离 d 。对任意一条方向角为 θ ，到原点的距离为 d 的并且经过 A 点的直线计数器 $a[d][\theta]$ 增 1。

$$\begin{aligned} \min X &= \min(\min X, x), \max X = \max(\max X, x) \\ \min Y &= \min(\min Y, y), \max Y = \max(\max Y, y) \end{aligned} \quad (4.4)$$

3) 扫描完图像中所有像素之后，找到数组 $a[M][N]$ 中最大元素，该元素所对应的坐标 (d, θ) 就是我们所要求的图中最长直线。该直线到原点的距离为 d ，方向角是 θ 。数组 b 中的元素 $b[d][\theta]$ 则记录了该直线的两端点坐标 $(\min X, \min Y)$ ， $(\max X, \max Y)$ ，将这两端点用区别于背景色和前景色的直线连接起来就标记了用 hough 变换找到的直线。

4.3.2 方形印章倾斜校正

利用 hough 变换检测直线固然有其简单性的特点。但是其计算量也是不可忽略的。尤其是在图像较大，前景点较多的情况下。如果前景点比较密集，有时甚至会得出错误的结果。观察方形印章可以看出，在其有一定厚度的边框部分，前景点比较密集，如果直接用 hough 变换很可能会产生错误的结果。

我们在第三章已经得到了方形印章的四个顶点的近似坐标，利用它们并结合 hough 变换就可以得到较为准确边缘直线。找到四条边缘直线就可以得到方形印章的四个顶点的准确坐标。观察图 4-6 可以得到印章外边缘顶点 A 的直线以边缘所在直线为界，在印章轮廓线外的直线其前景点个数远远小于边缘和跃过边缘经

过印章边框的前景点个数。根据这个特征，我们利用一种经过改进的 hough 变换来获取边缘。



图 4-6 过顶点 A 与边缘夹角为 θ 的直线

我们以寻找 A 点所在的右边缘为例，首先顺时针找到四个顶点中 A 点的下一个顶点 B。计算 AB 直线所在的方向角 θ ；则右边缘所在的直线必然是在 $[\theta - \text{threshold}, \theta + \text{threshold}]$ 这个小领域内变换，这个区域的大小一般 ≤ 10 ；这样 θ 的变换范围就比原 hough 变换的 0° 到 180° 之间缩小了近 90%，从而加快了寻找边缘的速度。我们在这里保存寻找到的边缘的端点和方向角。

一般来说只要精确到一条边缘即可，其他的边缘可以根据与该边缘的位置关系得到。找到该边缘的角度后，就可以根据几何关系计算出图像的倾斜角度。

4.4 圆形印章圆心的修正

从图 4-2 可以看出，由于外边缘轮廓有断裂和残缺的问题造成了选择了有误差的形心，而印章圆心的准确性直接关系到后来极坐标变换的正确性，从而影响到字符区域定位的准确性。因此我们必须对印章的圆心进行修正^[28]。

观察图 4-2，我们不难发现虽然其外轮廓有断裂和残缺，但其环形边框的内轮廓却是完整相连的。如果我们将背景点作为前景点，前景点作为背景点，即将原图像反色，如图 4-7 所示。我们实际上是在提取内轮廓之内环形背景区域的外轮廓。不过这个外轮廓之外有很多噪声，因此我们无法用扫描线提取边缘的方法。由轮廓的连续性这个特点，我们仍然选用链码跟踪算法来获得环形边框内边缘轮廓。我们的方法如下所示。

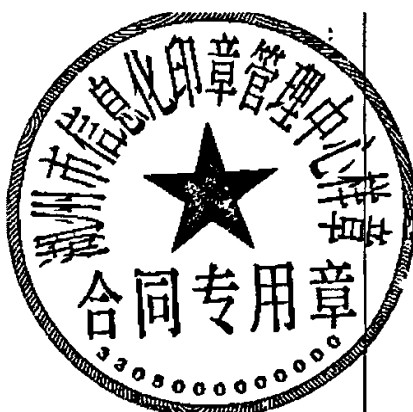


图 4-7 二值化图像反色所得图像

1) 根据第三章所提取的外轮廓点中选取一个点 A ；从该轮廓点开始进行水平扫描；如果从这个点开始某一方向出现前景点段和背景点段交替出现，并且相互的距离小于某一阈值 $threshold$ (通常 ≤ 2)，则继续向该方向搜索，直到出现背景点段大于阈值的情况为止。记录该长背景段出现之前的前景点坐标，该坐标即为环形边框内轮廓上的一点 $A1$ 。为了保证准确性，我们另外再选取与外轮廓上 A 点相邻的 B, C 两点，重复上述的扫描得到内轮廓上的点 $B1, C1$ 。综合分析 $A1, B1, C1$ 的位置，如果位置差异过大，很可能找到错误的内轮廓点。则可以重复上述方法，直到准确找到内轮廓上的一点位置。

2) 从该点开始，利用第三章中的链码跟踪算法找到环形边框内轮廓，只不过这次我们把背景点作为前景点，前景点作为背景点而已。



图 4-8 精确定位后的圆心位置

3) 依然是根据第三章中的求圆心和半径的方法得到修正后的圆心。由于圆形印章单从边缘轮廓和几何参数上是无法对其进行倾斜校正的,因此,我们将圆形印章的倾斜校正留在字符区域定位之后,根据字符区域的对称性和字符的方向性来对其进行有效的倾斜校正。本文中不研究圆形印章的倾斜校正。

通过内轮廓链码跟踪算法得到的圆心位置定位如图 4-8 所示。从图上可以看出该方法对圆心进行了有效的定位。

4.5 椭圆形印章的倾斜校正

4.5.1 精确椭圆印章形心

与圆形印章一样,椭圆形印章圆心的准确性直接关系到后来极坐标变换的正确性,从而影响到字符区域定位的准确性。修正椭圆形印章的圆形同样重要。

椭圆形印章与圆形印章的几何参数修正算法是一样的。我们采取和圆形印章一样的链码跟踪算法来取得椭圆印章边框内轮廓。根据椭圆的特点,修正椭圆的形心。找到形心之后,计算形心与内轮廓点的最大距离 max , 和最小距离 min , 从而得到椭圆长短轴 a, b 的精确比值 $k=a/b$ 。记录下椭圆形心与内轮廓上取得最大距离的点 A 的坐标 (x, y) 。

4.5.2 椭圆形印章倾斜校正方法

我们找到经过椭圆圆心 $O(x_1, y_1)$ 到内轮廓线上的最长距离的端点 $A(x, y)$ 。则 AO 就很可能是椭圆的长轴所在直线。则椭圆长轴的偏离角 ϕ 就可以按由公式(4.5) 得到。

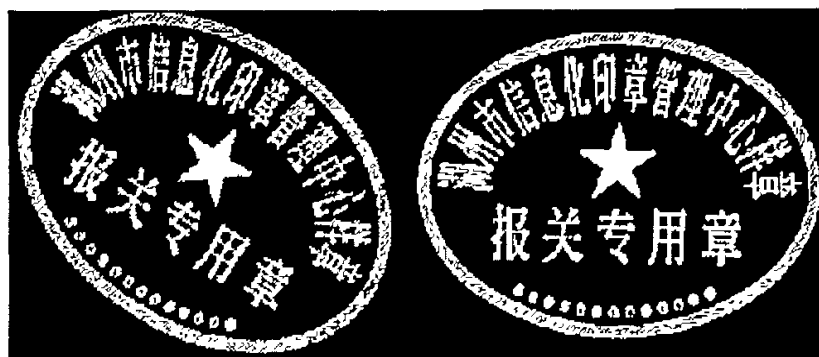


图 4-9 椭圆倾斜校正前和校正后的结果

$$\begin{cases} \phi = \arctan \frac{y - y_1}{x - x_1} \\ \phi < \pi/2 \end{cases} \quad (4.5)$$

根据偏转角即可对椭圆印章进行倾斜校正。椭圆倾斜校正的结果如图 4-9 所示。从图中可以看出，我们所得到的椭圆倾斜角度是正确的。

4.6 本章总结

本章对已经确定形状的印章参数进行精确性校正。在校正的基础上对方形和椭圆形印章进行倾斜校正。针对印章的形状不同，我们的侧重点也不同。

方形印章的倾斜性校正对我们后来进行的字符区域定位是非常重要的，因此尽管我们已经通过计算初步确定了四个顶点，我们为了准确进行倾斜性校正，仍然利用改进的霍夫变换，确定了至少一条外轮廓边的倾斜角度。

圆形和椭圆形印章侧重于形心定位的准确性。由图 4-2 所示，由于外轮廓的残缺和其它干扰的存在，我们找到的形心可能有较大的误差。这对于我们后来对圆形和椭圆形印章的字符区域定位是非常不利的，甚至无法对字符区域进行定位。因此我们对其形心进行了校正定位。圆形印章的倾斜校正将留到识别字符时进行。椭圆形印章的倾斜校正，则利用到了我们确定形心时的一些数据。简单而且快速，也产生了很好的结果。

第5章 字符区域特征提取

5.1 引言

印章图像中的重要内容包括文字和图像。对文字和图像的提取是印章特征提取中最重要的步骤。印章中的文字总是呈一排一排的形状，包括横排、上弧排和下弧排文字的排列。印章中的图像还有一些特定的几何图形，这些图形一般是印章所有者所自定义的。本章主要是研究印章中的文字区域特征提取。

5.2 印章图像的排字特征

一般文档中的文字都是采用垂直或水平排字方式。而印章中的文字排列受到印章形状的影响往往还需要用到弧形排字方式，包括圆或椭圆排字方式。印章中的水平排字方式往往采取普通文档中的等间距排版方法。所以本节主要讨论圆或椭圆的排字方式。

5.2.1 印章的圆形排字特征

为了达到美观的效果，圆形印章图像中字符文字的设计往往采用圆形排字方法。设计圆形排字方法的步骤如下：

- 1) 统计印文文字属性，如文字内容，字符个数。
- 2) 设计印文的排列属性，如上（下）弧形，起始角度，中心距离，字体，字间距等等。
- 3) 计算印文的分布角度。从印章圆心做等角半径旋转，从而得到字符输出角度与半径角度一致，并且中心距离相等的圆弧形均匀排字方案^[2]。

5.2.2 印章的椭圆形排字特征

圆形印章采取的以印章圆心作等角半径旋转从而得到均匀弧形的排字方案，对于椭圆形印章却无法得到相应的以椭圆形均匀弧形文字排列。有人提出以中心距弦长为基础的迭代平衡算法^[2]，根据该算法，可以得到椭圆的弧形文字排列的特点。

5.2.2.1 中心距弦长的定义

中心距弦长的定义：如图5-1所示，中心射线 OP 与椭圆边界交于 P ，直线 AP

是经过 P 点的椭圆边缘的切线；中心射线 OQ 与椭圆边界交于 Q ，直线 BQ 是经过 Q 点的椭圆边缘的切线。 AP 与 BQ 线段的长度为 r ；则 AB 则为射线 OP 与 OQ 之间的弦心距。 AB 的长度就是 OP 与 OQ 之间距离椭圆边缘距离为 r 的中心线弦长。

5.2.2.2 椭圆弧形字符排字方案

椭圆印章弧形字符排字方案：选定起始角度，按照离椭圆边缘距离为定制 r 的中心距弦长为一个字宽的距离增加中心射线角度。以中心距弦长的中垂线方向安放相应字符。这样就可以得到均匀的椭圆印章弧形字符排列。

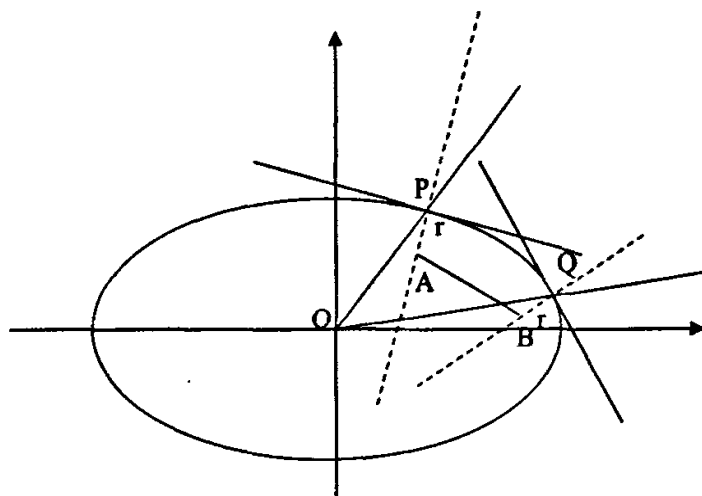


图 5-1 椭圆中心距弦长示意图

5.3 常用字符区域定位算法

5.3.1 边缘检测与扫描线相结合的字符区域定位算法

这种算法利用了文档中字符区域中字符排列的连续特性。字符区域通常有连续多个字符，设为 N (设 $N > 5$)，而且字符之间的距离也在一定距离内。算法中定义从前景点到背景点或者从背景点到前景点的变换为一个跳变。字符区域相对于非字符区域有这样的特点：跳变多，而且在一定距离内的跳变次数大于一定阈值。这个值通常为 $2N$ 以上，因为字符区有 N 个字符，每个字符有两个以上跳变。该算法描述如下。

1) 按从上到下的顺序，对图像的每一行进行从左向右的扫描。到跳变点记录下该点的当前位置，如果某行有连续 m 个跳变点，并且 m 大于阈值，并且前一

个跳变点和后一个跳变点的距离在一定范围内，就记录下起始点和终止点位置。

2) 如果连续有十行以上这样的跳变点，并且相邻上下行的起始点和终止点相邻。就认为该区域是字符预选区域。

这种方法适宜于在背景单一的情况下的横排字符区域的定位，而对于有复杂背景的图像则是不适合的。

5.3.2 DCT 压缩域快速定位算法

这种算法首先对图像作 8×8 DCT变换^[14]，即进行 8×8 离散余弦变换。通过研究图像中字符和非字符 8×8 块DCT系数的特点，可以看出，字符区的DCT系数具有更为丰富的中高频分量，因为字符区往往具有很多边缘；同时，字符区往往还具有较大的水平/垂直，以及斜向分量。综合考虑DCT系数的方向和频率因素，有人提出了加权频率^[14](*WF*, weighted frequency)特征，利用它来进行字符块和非字符块的分类。

分类方法是：考虑每个 8×8 块的64个DCT系数在块中的位置，对这些系数的绝对值进行位置加权求和。通过对不同方向通道中的高频能量进行增强，使字符块的特征更为鲜明，然后设定一个阈值来确定每一个块的字符或非字符属性。加权频率*WF*的计算公式如5.1所示。式中的 $c(i, j)$ 为 8×8 块中第*i*行，第*j*列的DCT系数；*i, j*的范围取0~7。

$$WF = \sum_{i,j=0}^7 i \times j \times |c(i, j)| \quad (5.1)$$

在我们进行分类时，如果图像中字符块和背景之间的对比度较低，*WF*值会比较小；另外，高频信息丰富的非字符背景常常具有较大的*WF*值。因此，如果采用固定阈值，则不利于低对比度字符的提取，同时也容易将高频信息丰富的背景误认为字符区域。为此采用自适应阈值法进行分类，如(5.2)所示。

$$WF_threshold = aver_WF + (max_WF - min_WF) \times K1 \quad (5.2)$$

其中，*aver_WF*为整幅图像的平均*WF*值，*max_WF*，*min_WF*分别为整幅图像的最大、最小*WF*值，*k1*为可变参数，它是根据经验值而设定的。

这样，图像中*WF*值比较大的字符区被保留下来，而其它大片的背景区域则被剔除出了图像，从而定位出了字符区域。

这种方法显然适合于在背景比较复杂的情况下确定字符区域，尤其还可以用于有多字符区域的图像。

5.3.3 投影分割定位算法

对于经过二值化,平滑,去噪后的字符区域图像,可以首先对其进行水平投影,然后对投影值进行分析,通过投影直方图的峰值和凹谷确定出水平基线;再通过对水平基线之间的图像区域进行垂直投影^[24],确定出垂直基线。这样就可以初步定位出字符区。这种方法是字符定位和分割的常用方法。基线的产生通常依据下面的规则。

将投影区中过小的投影值置为零,然后当相邻投影值有非0到0的跳变时,即认为在非零投影值处存在一条基线。

在背景复杂的图像中往往会存在一些与字符区相连的噪声区,例如背景纹理,物体的边缘等,所以在对字符区进行粗略分割之后还需要进一步细分割,即对每一个已经初始定位的字符框,再次进行二次水平、垂直投影和分割。为了将字符区准确地分割出来,在确定水平基线时根据最大、最小投影值,设定一个自适应分割阈值作为分割条件。当相邻投影值之差大于 $division_threshold_1$, 或者投影值高于 $division_threshold_2$ 时,即认为该处需要设定基线,其中 $division_threshold_1$ 和 $division_threshold_2$ 的设定如下式(5.3)所示,式中的 $k1$ 、 $k2$ 为经验值。

$$\begin{aligned} division_threshold_1 &= (\max_pro - \min_pro) * k1 \\ division_threshold_2 &= classification_map_width * k2 \end{aligned} \quad (5.3)$$

式(5.3)中的 \max_Pro , \min_Pro 分别为该字符框中的水平最大和最小投影值;而 $classification_map_width$ 为图像的宽度。之所以采用这种细分割模式,是因为在复杂背景的字符图像中常常存在着很多噪声区。如果把这些噪声区和字符区一起分割定位出来,则易造成定位出的字符区和非字符区的对比不明显的问题出现。这不仅影响到后面假字符区的去除工作,同时也容易将包含噪声的字符区作为假字符区滤除。采用二次投影分割方法可以减小字符框中的非字符背景,使定位结果矩形框更紧凑地包围字符^[14]。

5.4 方形印章字符区域定位算法

从方形印章经过二值化和旋转校正之后的图像可以看出,除了四周有一定宽度的边框,文字区域部分就像是普通文档的文字部分一样,呈文字行和背景行相互交替出现的情况。所以我们可以首先将周围的边框除去,再利用水平投影法和垂直投影法确定文字区域。

5.4.1 除边框方法

仔细研究图 2-1 中的方形印章可以得到下面的结论，边框区域的前景像素点比字符区域的密集地多。根据这个结论，我们如果对印章图像进行水平或竖直投影，可以在方形印章的边框区取得边框宽度的连续峰值。算法如下：

1) 对印章图像作竖直投影。投影得到的最高峰值 $max_threshold$ 。和除零以外的最低峰值 $min_threshold$ 。取一个参数 $k(k$ 一般 $<1/2$)，求一个阈值 $threshold$ ：

$$threshold = k * (max_threshold + min_threshold) \quad (5.4)$$

该阈值用来判定是否是图像边框。

2) 从印章图像左边开始向右扫描投影区域，如果投影值 $> threshold$ ，并且符合这样条件的区域是连续不间断的(区域宽度 > 10)，则判断为边框区域，直到投影值 $< threshold$ 为止。记录下这个区域的起始 $x1$ 和终止水平位置 $x2$ 。

3) 从印章图像右边开始向左扫描投影区域，如果投影值 $> threshold$ ，并且符合这样条件的区域是连续不间断的(区域宽度 > 10)，则判断为边框区域，直到投影值 $< threshold$ 为止。记录下这个区域的起始 $x3$ 和终止水平位置 $x4$ 。

4) 对印章图像水平位置 $x2$ 和 $x4$ 之间的区域作水平投影。

5) 由于已经去除了左右边框，则上下边框与文字区域之间分别有背景区域隔离。从上到下扫描印章图像投影区域，直到遇到第一个连续不间断非 0 投影区域(区域宽度 > 10)，继续扫描直到扫描到 0 投影值位置。记录这个区域的起始位置 $y1$ 和终止位置 $y2$ 。

6) 从下到上扫描印章图像投影区域，直到遇到第一个连续不间断非 0 投影区域(区域宽度 > 10)，继续扫描直到扫描到 0 投影值位置。记录这个区域的起始位置 $y3$ 和终止位置 $y4$ 。

7) 通过上述的一系列处理，则可得到除去边框的文字区域，即以 $(x2, y2)$ 和 $(x4, y4)$ 之间的线段为对角线的矩形区域。接下来就可以对这个矩形区域内的字符进行定位。

5.4.2 字符区域定位

去除了边框的矩形区域与一般文档无异。并且已经得到了文字区域的水平坐标。只需要垂直坐标即可。则采取水平投影方式。对每一个连续非 0 投影值区域记录其起始和终止位置。最后得到方形印章的所有文字区域的对角坐标。

5.5 圆形印章字符区域定位算法

圆形印章有关于圆心的对称的特点。因此，要提取其字符区域，必须首先对其进行极坐标变换。为了适应离散图像，这里提出了一种自适应的极坐标变换方法。

5.5.1 自适应的极坐标变换方法

对一个离散的圆来说，单纯的几何学极坐标转换方法不太合适。比如说在 $r=1$ 的情况，在总共只能有 $0,90,180,270$ 这几个角度。而在 $r=1023$ 的情况，几乎 1 度可以有 $1023/90=11$ 个像素点。如果我们依然按照常规的直角坐标向极坐标转换的方法来处理离散圆，则可能会丢失或者使得很多像素信息重合。如果进行投影转换则很可能产生很大的误差，不利于图像的分割。这里提出一种针对离散圆的自适应的从直角坐标转换为极坐标的方法。基本思想是：先利用改进的 Bresenham 算法对圆进行极坐标变换。在两个像素之间多于 1 度时，进行插值，务必使每一度中有一个像素；在两个像素都属于一度时，根据一度中所有像素的亮度信息的平均值与阈值的比较来确定代表该度的像素的亮度值。具体算法如图 5-2 所示：

在该算法的第 1 行，数组 $pE\theta$ [i] 记录角度 i 出现的次数。 $pE\theta$ [i] 则记录角度 i 是前景点的次数。第 10 行中，根据误差项 d 的判断，决定非最大位移方向上是走还是不走。在第 15 行后的处理过程同样适用于关于圆心对称，半径为 r 的其它七个点。

```

1. 初始化数组 pEtheta[0-359], ptheta[0-359];
2. y=r, x=0; d=1-r; 圆心为 (cx, cy);
3. WHILE x<=y DO
4.   theta=asin(x/(x*x+y*y))/pi*180;
5.   pEtheta[90-theta]++;
6.   IF (cx+x, cy+y) 为前景点 Then
7.     ptheta[90-theta]++;
8.   END IF
9.   IF d<0 THEN
10.    d+=2*x+3;
11.   ELSE d+=2*(x-y)+5;
12.    y--;
13. END IF

```

```
14.     x++
15. END WHILE
16. FOR theta=0 TO 360 DO
17.     IF pEtheta[theta]<>0 THEN
18.         IF ptheta[theta]<>0 THEN
19.             lpZ(theta,r)=1;
20.         END IF
21.         Theta++;
22.     ELSE
23.         Temp=theta;
24.         Theta++;
25.         WHILE pEtheta[theta]==0 AND theta<360 DO
26.             Theta++;
27.         END WHILE
28.         IF ptheta[(temp-1+360)%360]&&ptheta[theta]
29.         DO
30.             FOR x=temp To theta BY 1 DO
31.                 LpZ(x,r)=1;
32.             END FOR
33.         END IF
34. END FOR
```

图 5-2 圆形的自适应的极坐标变换算法

图 5-3 显示了利用一般极坐标变换方法和自适应的极坐标变换方法后进行以圆心的端点的中心射线结果。图 5-3 中的(a)是利用一般极坐标变换方法对印章图像进行极坐标变换之后对五角星字符进行环形投影后的结果。图 5-3 中的 (b) 则是利用自适应的极坐标变换后的结果。通过比较可以看出, 尽管五角星是一个连通的字符, 而一般极坐标变换方法却把它分为了五个字符。这将给我们后来的字符分割和识别造成很大的困扰。而自适应的极坐标变换方法则不会出现上述的问题。

5.5.2 确定字符区域

经过观察发现，圆形印章以圆心为中心，其字符区域呈现环形分布。因此我们首先对它进行环形投影变换，找出可能的字符区域，投影结果如图 5-4 所示。从图中可以看到投影区域清晰地分为三个部分，分别代表圆心位置的图形，中间的文字部分，以及边框图像。记录每个投影区域的起始和终止半径长度。

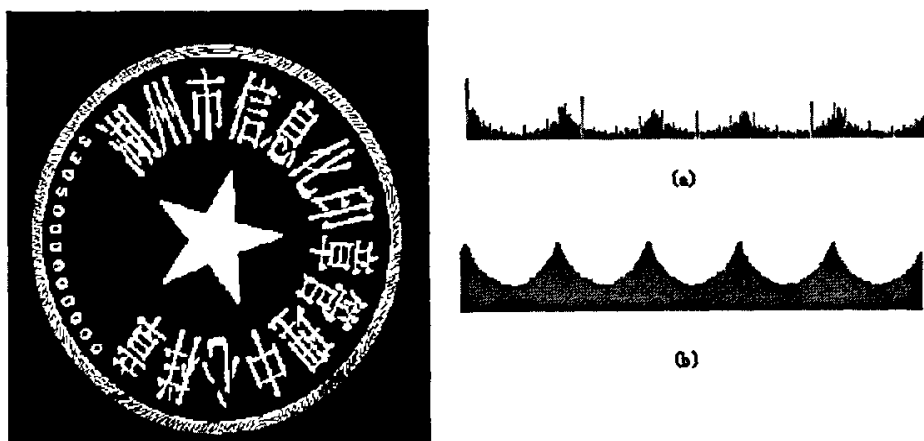


图 5-3 一般极坐标变换和自适应极坐标变换方法的结果对比



图 5-4 环形投影结果

接着对除了边框以外的字符区域进行从以圆心的端点的中心射线投影，从图 5-5 可以看出中间环形区域靠近圆心的圆弧字符部分有字符粘连在一起的情况。为了避免在投影时产生误差，我们选择从靠近圆心字符区域的起始半径 $r1$ 加上某一个阈值 $threshold$ 作为投影时的起始半径。这个 $threshold$ 可以依靠经验值求得，如 $threshold$ 可以等于 $(终止半径 r2 - 起始半径 r1) / 5$ ，可以得到如图 5-5 的投影效果。

从图 5-5 上可以看到，每一个字符都对应着一个投影区。图 5-5 中汉字区较大，而且每个汉字占据的扇形角度大小相同，而两个汉字之间的空白区域角度远小于汉字所占角度。而每个数字所占扇形角度则远小于汉字，并且数字之间的空白与字宽差不多。根据这个事实来确定汉字的区域和数字的区域。

从图 5-5 的投影区可以看到，有一个汉字的投影出现了断裂。为次当出现不能确定分割字符的时候，我们将进行二次投影^[6]和直方图修正，直到确定相应字符区域的扇形角度为止。最后根据相似性，合并字符区域。得到字符的扇形角度之后，对这个角度之内的区域进行环形投影，根据环形投影的投影特征，得到分割结果。



图 5-5 字符区域中心射线方向投影结果

对图 5-5 中的印章中心环上的字符分割结果如表 5-1 所示。从表上可以看出得到的字符区域误差很小，说明定位是准确的。

表 5-1 圆形字符分割角度结果

字符区域	起始角度	终止角度	字符与空白区域比值
区域 1	134	234	<3
区域 2	235	(133+360)	>3

5.6 椭圆形印章字符区域定位算法

椭圆与圆一样同样也要用到极坐标变换。由于椭圆与圆一样，在离椭圆圆心较近的圆弧区域，如短轴长为 1 的椭圆弧可能只有四个点，其角度也只有 $0, 90, 180, 270$ 这几个角度。当我们进行中心射线投影时，可能会造成误差，因此也需要用到自适应的极坐标变换方法。由于椭圆自身结构的原因，其极坐标方法与圆的极坐标变换方法又有所不同。

5.6.1 椭圆的自适应极坐标变换方法

椭圆的极坐标变换仍然选用的是改进的 Bresenham 算法。这里，我们不但要讲每个点的坐标由直角坐标转换为极坐标，还要标记属于同一个椭圆环的点。因此我们首先定义一个结构 EPOINT;

```
Typedef struct
{
    Int theta;//方向角
    int r;//点离形心的距离
}EPOINT;
```

对 90 度距离椭圆圆心距离为 m (m 在 1 到 b 之间) 的椭圆弧，求其极坐标的具体算法如图 5-6 所示。图 5-6 中第 1 行分配一个以 EPOINT 为元素的二维数组 $lpEZ$ ，其第一维为 360 ，第二维为短轴长度 b 。数组 $pEtheta[i]$ 记录角度 i 出现的次数。 $ptheta[i]$ 则记录角度 i 是前景点的次数。 $Pr[i]$ 则记录该角度圆弧上的点到椭圆圆心的距离。图 5-6 中第 3 行的 a 表示椭圆的短轴长度； b 表示椭圆的长轴长度。

通过自适应的极坐标变换之后，椭圆中的五角形连通字符同样不会出现中心射线投影断裂而影响字符分割的问题。

1. 初始化数组 $pEtheta[0-359]$ ， $ptheta[0-359]$ 所有元素为 0;
2. 初始化 $y=m, x=0$;
3. 初始化差值 $d=b*b+a*a(0.25-b)$; 椭圆圆心为 (cx, cy) ;
4. WHILE $x \leq a$ DO
5. $R = x*x+y*y$;
6. $Theta = \text{asin}(x/R) / \text{pi} * 180$;
7. $Pr[90-theta] = R$;
8. $pEtheta[90-theta]++$;


```
9.   IF (cx+x, cy+y)为前景点 DO
10.     pTheta[90-theta]++;
11.   END IF
12.  x++;
13.   IF d<0 THEN
14.     d+=2*x+3;
15.   ELSE d+=2*(x-y)+5;
16.     y--;
17.   END IF
18. END WHILE
19. FOR theta=0 TO 360 DO
20.   IF pTheta[theta]<>0 THEN
21.     IF pTheta[theta]<>0 THEN
22.       lpEZ(theta, r)=EPOINT(theta, m);
23.     END IF
24.     Theta++;
25.   ELSE
26.     Temp=theta;
27.     Theta++;
28.     WHILE pTheta[theta]==0 AND theta<360 DO
29.       Theta++;
30.     END WHILE
31.     IF pTheta[(temp-1+360)%360] &&pTheta[theta]
32.     THEN
33.       FOR x=temp To theta BY 1 DO
34.         LpERZ(theta, m)= EPOINT(theta, r);
35.       END FOR
36.     END IF
37. END FOR
```

图 5-6 椭圆的极坐标变换方法

5.6.2 确定字符区域

经过观察发现，椭圆形印章以椭圆圆心为中心，其中心和边框都是呈环形分布的。我们首先对它进行椭圆环形投影变换。通过环形变换的结果将椭圆中心的图像区域和边框去除。然后再剩下的区域中找出可能的字符区域。

椭圆形印章中的字符区域的排字方式可分为两种：一种是椭圆弧型的，其结构已经在本章的前面阐述过了；另一种则是如一般文档中一样是横向排列的。

5.6.2.1 定位椭圆中的横向排列字符区域

对于横向排列的字符区域，定位它们的关键是要确定其上下和左右边界。我们利用边缘检测和扫描线相结合的字符区域检测和定位算法的原理来对该区域进行定位。我们已经对椭圆印章进行了倾斜校正，假定所要定位的椭圆的长轴已经在水平线上。

对椭圆印章进行从上到下，从左到右扫描，记录下每一行中每一段的背景区域和前景区域。如果从某一行开始连续发生的跳变很多，而且这些跳变都发生在一个固定的左右区域，即横坐标在 min_x 和 max_x 之间，并且 min_x 和 max_x 与这之外的前景点之间背景区域的长度比发生很多跳变得区域中的相邻前景点之间的背景区域要大，则可以初步确定该横向字符的左右区域。继续扫描直到该区域之间的跳变变为 0 或者较少到原来的 $1/N(N>10)$ 为止。记录下起始和终止的纵坐标 min_y 和 max_y ，则它们为横向排列字符区域的上下边界的纵坐标。则以 (min_x, min_y) 和 (max_x, max_y) 为对角点的方形区域即为横向排列字符区域所在区域。

最后我们还可以用垂直投影来验证我们所检测到的字符区域的正确性。如果直方图的投影区和空白区交替出现而且投影区宽度差不多，则可以证明横向排列字符区域定位成功。

5.6.2.2 定位椭圆中的弧形排列字符区域

定位了椭圆字符区域中横向排列的字符区域之后，剩下的就是弧形排列字符区域。椭圆中弧型排列的字符区域主要是根据中心距弦长等于字宽的原理分布的。根据图 5-1 中中心距弦长的定义，我们可以设计下列方法来定位这些区域。

- 1) 首先要对去除了中心字符区域、边框、横向排列的字符区域之后的区域进行自适应的极坐标变换。

- 2) 根据极坐标变换的结果进行中心射线投影。从中心射线投影的结果，可以初步判断出字符相似的区域所占的角度。

3) 判断出所占角度之后, 针对该角度进行环形投影, 记录下环形区域短轴方向上到椭圆圆心的距离 r_1 和 r_2 。

4) 从到椭圆圆心较远的初始定位的距离一端找到一个起始前景点。取离这个前景点一定距离的椭圆圆弧上的背景点。该背景点所在中心射线角度为 θ 。在经过该背景点的 $[\theta-h, \theta+h]$ ($h < 10$) 角度区域对每一个角度进行前景点值统计。如果某一角度的前景点的统计值为 0, 说明找到椭圆弧排列字符之间的一条分割线。如果没有前景点统计值为 0 的直线, 则换一个椭圆弧上的点继续寻找, 直到找到为止。相似字符拥有相同的字宽, 只要找到相邻的两条这样的分割线就可以了。这样就可以定位出椭圆弧排列字符的区域。

5.7 本章总结

本章研究了印章中字符区域的定位方法, 主要研究印章中常见的方形、圆形和椭圆形印章。首先对印章中主要的字符排列方式进行了研究和阐述, 尤其是对椭圆弧字符排列方式的原理进行了详细的介绍。接着研究了当前主要的字符区域定位的算法。最后分别对方形、圆形和椭圆形印章的字符区域定位方法进行了研究和说明。分别针对其形状和几何特点以及字符分布的特点采取了相适应的区域定位和字符分割方法。

对于方形印章主要用了投影分割定位算法。首先去除边框, 再对字符区域进行定位。对于圆形和椭圆形印章首先用基于改进的 Bresenham 算法基础上的自适应极坐标变换方法进行极坐标变换。并将普通极坐标变换和自适应极坐标变换的变换结果进行了比较。通过比较可以看出, 自适应的极坐标变换方法不会在接近形心的区域对字符的分割和识别产生误差。接着针对圆形印章的字符排列特点, 采取环形投影和中心射线投影相结合的方法确定字符区域, 并对得到的直方图进行直方图修正得到正确的分割结果。最后椭圆形字符的排列特点, 采取环形投影和中心距弦长步长变换寻找字宽的方法来确定相似椭圆排字方式的字符区域。而对于椭圆形字符中矩形区域则采用了扫描线投影字符区域定位方法。字符区域定位的成功与否将关系到我们以后的字符识别工作的正确性。

第6章 总结与展望

6.1 总结

本文研究印章图像的特征提取算法。本文分模块、分步骤地研究了多种改进算法，研究的主要成果为：

1) 针对印章的特点，研究适合印章图像二值化和去噪处理的算法。利用这些算法得到的二值化图像非常有利于我们进行轮廓提取、几何特征提取和字符区域的提取和识别。

2) 在印章形状的识别过程中，论文提出了结合链码跟踪法和余弦法的轮廓端点提取算法，并进一步根据顶点的个数和几何位置确定方形、圆形、椭圆形和三角形印章的形状。

3) 针对圆形和椭圆形印章因为外轮廓残缺从而导致形心无法正确找到的情况，提出利用链码跟踪算法提取内轮廓的方法来校正形心，从而为后来的极坐标变换提供正确的保证。

4) 针对椭圆和圆形印章提出了一种基于改进的 Bresenham 算法的极坐标变换方法。该方法的原理是对半径过小时，对缺失角度进行插值来实现的。

5) 针对不同形状的印章，研究了几种字符区域定位和分割方法，采用适合于该形状印章的字符定位方法，并且效果显著。

6.2 进一步的工作

由于时间仓促再加上印章样本的不足，本文的工作还有不足之处，进一步的工作主要集中在以下几个方面：

1) 多收集一些印章样本，改进算法。主要从提高特征提取的正确率和速度。加快印章特征提取和识别的速度，以适应网络或者实时交互的需要。

2) 研究新算法。现有的算法是针对形状规则的方形、圆形和椭圆形印章，而事实上可能还有其它的较不常见的形状。因此，必须继续研究可适用于各种类型的印章特征提取算法。

3) 当提取到印章字符区域后，我们还将对字符区域中的汉字或者数字进行识别后存储，这将用到很多模式识别^[29]等字符识别的技术^[30]。因为印章的特征提取还不能自成为一个被市场所接受的通用系统，只有进一步进行印章中字符的识

别和存储，才能够扩展为一个独立的有实用性的系统。

参考文献

- [1] 张小绵, 柳松等. 印章网络管理系统的设计与实现. 株洲工学院学报, 2002, 60(4):46-47
- [2] 张小绵, 古元等. 面向对象的印章排版和识别. 中山大学学报论丛, 2000, 20(5):59-60
- [3] 周琨, 张大力. 印章鉴定系统的图像处理研究. 计算机工程与应用, 2002, 38(20):198-200
- [4] 赵永涛, 李志敏等. 印章识别中的预处理研究. 仪器仪表学报, 2004, 25(4):402-410
- [5] 杨峰, 周洪利. 印鉴自动识别系统中模式识别方法的研究. 现代计算机, 2003, 155(1):6-9
- [6] 陈新燕, 李卫东等. 基于边缘和模板匹配的印鉴自动鉴别方法. 上海交通大学学报, 1998, 32(4):44-48
- [7] Q.Hu.An Automatic Seal Imprint Verification Approach. Pattern Recognition.1995, 28(8):1251-1266
- [8] 陈志云, 李志敏等. 基于Hough变换的印鉴差值匹配识别. 重庆大学学报(自然科学版), 2005, 29(2): 49-50
- [9] 张玲, 张永军等. 鉴印系统中印章几何参数的计算. 信息与电子工程, 2006, 4(3):197-199.
- [10] Milan Sonka, Vaclav Hlavac, Roger Boyle著, 艾海舟、武勃等译. 图像处理、分析与机器视觉(第二版). 北京: 人民邮电出版社, 2003
- [11] 赵海燕, 马驹良等. 基于最小面积法进行的车牌字符分割. 长春理工大学学报, 2006, 29(3):47-48
- [12] 张玉林, 张永华. 基于自组织映射神经网络的粘连字符分割. 山东科技大学学报, 2004, 25(2):117-120
- [13] 王建宇, 张峰等. 利用小波变换和K均值聚类实现字幕区域分割. 计算机辅助设计与图形学学报, 2006, 18(10):1508-1510
- [14] 孙慧平, 刘党辉等. 一种新的DCT压缩域字符快速定位算法. 测控技术, 2005, 24(5): 49-50
- [15] D.H.Ballar. Generalizing the Hough transform to detect arbitrary shapes. Pattern

- Recognition, 1981, 13(2):111-122.
- [16] J.Goutsians, J.Henk,Hei Imans. Nonlinear Multiresolution Signal Decomposition Schemes-Part: Morphological Wavelet. IEEE Transactions on Image Processing, 2000, 9(11):1997-1913
- [17] 张立明. 人工神经网络的模型及其应用. 上海: 复旦大学出版社, 1993
- [18] Y.W. Lei, K.C. Wong. Ellipse Detection Based on Symmetry. Pattern Recognition Letters, 1999, 20(1):41-47
- [19] 叶家鸣. 彩色城市交通地图道路信息的识别与提取[硕士论文]. 北京: 中国科学技术大学, 2003
- [20] 叶晨洲, 杨杰等. 车辆牌照字符识别, 上海交通大学学报(增刊), 2000, 34(5):672-675
- [21] 吕俊哲. 图像二值化算法研究及其实现. 科技情报开发与经济, 2004, 14(12):266
- [22] J. N. Kapur, P. K. Sahoo and A.K.C. Wong. A New Method for Gray-Level Picture Thresholding Using the Entropy of the Histogram. Computer Vision, Graphics and Image Processing, 1985, 29(3):273-285
- [23] 杨玲. 字符图像混合二值化方法. 辽宁省交通高等专科学校学报, 2006, 8(4):35-37
- [24] N. Otsu. A Threshold Selection Method From Gray-Level Histograms. IEEE Transactions on Systems, Man, and Cybernetics, 1979, 9 (1):62-66,
- [25] J.R.Parker. Gray Level Thresholding in Badly Illuminated Images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1991, 13(8):813-819
- [26] 胡钟山, 娄震等. 文档处理中消除线噪声的研究. 计算机研究与发展, 1999, 36(8):992-995
- [27] 刘重庆, 曾东武等. 在印鉴识别中的配准方法. 上海交通大学学报, 1998, 32 (9):46-48.
- [28] 王洪建, 刘波等. 圆形印章中心定位算法的研究. 仪器仪表学报, 2006, 27(6):2256-2258
- [29] 边肇其. 模式识别. 北京: 清华大学出版社, 1994
- [30] 胡家忠. 微机应用丛书: 计算机文字识别技术. 北京: 气象出版社, 1994
- [31] 长沙睿思科技开发有限公司网站. <http://www.ggzhzh.bankbuy.cn>.

致谢

时间如白驹过隙，我在浙江大学攻读硕士学位的生活就要结束了，欣喜之余也充满着颇多感动。在这两年的时间里，我的收获很大、进步很快，这些都与身边一直关心、帮助、支持我的老师、同学和朋友们是分不开的，是他们的关心和帮助才使我能顺利完成学业，使我度过这段美好的、令人难忘的时光。

感谢实验室里辛勤培育我的导师张引副教授，是她的精心指导和严格要求才使我能够顺利地完成本论文的研究工作。实验室里的另外两位老师叶修梓教授和张三元教授也给了我很大帮助。这三位老师严谨的治学态度、渊博的知识、精益求精的工作作风，永远是我今后学习的楷模。在整个硕士阶段的学习中，他们不仅传授我知识，在学习上给予我指导和关心，更以师长风范教育我如何为人处事。在此谨向三位老师致以由衷的感谢和崇高的敬意！

感谢徐舒畅、褚一平等实验室里的博士师兄们，你们平日里为我解答难题，把宝贵的经验传授给我，使我很快的融入到理论研究工作中，指导我查资料、写论文，度过了许多难关。祝各位师兄今后多出成果，研究课题不断突破、不断创新。

感谢实验室同级的同学们，大家一起学习、互相帮助、共同进步，研究生期间因为有了你们才使我们的学习、生活丰富多彩。感谢我的父母，他们无私的爱和奉献是对我最大的支持，他们是我不断进取的动力和支柱。

曾锐

2007年5月于浙江大学CGIM实验室