

東南大學

硕士学位论文

指纹分类和识别算法的研究与实现

专业名称: 模式识别与智能系统

研究生姓名: 尹婉琳

导师姓名: 叶桦 教授



RESEARCH AND IMPLEMENTATION OF FINGERPRINT CLASSIFICATION AND IDENTIFICATION

A Thesis Submitted to

Southeast University

For the Academic Degree of Master of Engineering

BY

Yin Wanlin

Supervised by

Professor Ye Hua

School of Automation

Southeast University

April 2015

东南大学学位论文独创性声明

本人声明所呈交的学位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经发表或撰写过的研究成果，也不包含为获得东南大学或其它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。

研究生签名： 尹晓林

日 期： 2015.4.8

东南大学学位论文使用授权声明

东南大学、中国科学技术信息研究所、国家图书馆有权保留本人所送交学位论文的复印件和电子文档，可以采用影印、缩印或其他复制手段保存论文。本人电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文和在技术保护期限内的论文外，允许论文被查阅和借阅，可以公布（包括以电子信息形式刊登）论文的全部内容或中、英文摘要等部分内容。论文的公布（包括以电子信息形式刊登）授权东南大学研究生院办理。

研究生签名： 尹晓林

导 师 签 名： 叶 桦

日 期： 2015.4.8

摘要

随着互联网时代的到来,生物特征识别技术受到了社会各界的广泛关注,其中指纹识别技术一直都是国内外研究的热点和难点。越来越多的自动指纹识别产品被开发应用于人们的生活,提供身份鉴别及安全防护的功能。尽管目前在理论研究上已经取得了显著的成绩,但基于大规模指纹数据库的自动指纹识别系统在识别准确率和处理速度上都亟待进一步的提高。本文结合目前数字图像处理技术和模式识别技术的发展,深入研究了指纹识别的关键算法,主要的工作和创新总结如下:

1. 指纹图像预处理方面。在分析和研究了前人成果的基础上,实现了指纹预处理算法中的关键步骤,并着重改进了直方图均衡化、方向场计算和图像增强方法。显著增强了脊线和谷线的对比度,保留了奇异区的方向变化规律,提高了指纹图像的清晰度,为特征提取等操作的顺利进行奠定了基础。

2. 指纹图像分类方面。针对由于采集仪传感器面积小而无法提取三角点这一分类特征的不足,研究并提出了一种包括分类预处理和多级分类两个步骤的指纹分类方法。在分类预处理阶段,根据中心点附近的全局特征在指纹摆正时具有可分类性的特点,首先采用拟合中位线法和最小外接椭圆法确定指纹的倾斜方向,再通过仿射变换对指纹图像进行旋转校正。在此基础上,设计了一套具有粗定位、快提取、基于人类经验去除伪中心点的中心点提取算法。在多级分类阶段,建立了一套基于中心点数目、方向、斗型螺径、箕型中心拱形方向分布特征的三级指纹分类算法,实现了大规模指纹数据库的自动分类。

3. 指纹图像匹配方面。基于传统点模式指纹匹配算法计算较为耗时的缺点,提出了一种包括初匹配和全局匹配的两级指纹匹配算法。初匹配所比对的特征向量是由外接全等三角形和邻近点构成的三角形这两种子结构所组成,如果匹配成功则直接退出,否则仍能为全局匹配剔除掉部分不符合的样本图像和参考点对,达到减少全局匹配计算量的目的。实验结果表明,在全局匹配前增加初匹配能够大幅度降低匹配时间,并提高识别准确率。

4. 为了验证自动指纹识别相关的算法的可行性和实用性,结合设计模式思想和数据库相关技术,基于 VS2010 和 MYSQL 搭建了自动指纹识别系统软件平台。该软件整合了多个关键算法,可用于对指纹识别算法进行测试和研究。

大量实验数据结果表明,本文提出的预处理改进算法能够获得清晰的指纹图像,多级分类方法也具有较高的分类准确性,最终的二级匹配算法处理效果又快又好。

关键词: 指纹识别, 大规模指纹库, 指纹预处理, 旋转校正, 中心点提取, 指纹三级分类, 指纹二级匹配

Abstract

With the advent of the Internet Era, Biometric identification has attracted lots of attention and made great progress since the 80s. Fingerprint, one of the focuses in the personal identification field, is the hotspot and difficulty to researchers at home and abroad for a long time. Currently, an increasing number of automatic fingerprint identification systems (AFIS) have been developed and expanded in people's daily life, providing identity authentication and security protection. Although the theoretical research has made significant achievements, AFIS based on a large-scale fingerprint database is still to be further improved on accuracy and efficiency. Combined with the current development of digital image processing and pattern recognition techniques, this thesis researched several key algorithms based on large-scale fingerprint database, mainly focused on the optimization of the fingerprint image preprocessing, fingerprint classification and fingerprint identification. The main contributions of the dissertation are as following:

1. Due to different kinds of noises, the fingerprint image prerocessing is very important to improve the quality of the collected images. On the basis of the analysis of current fingerprint identification technology dissertations, this thesis realizes the key steps of the fingerprmt image preprocessing. In the regard, a series of algorithm modifications about histogram equalization, orientation field calculation and image enhancement is proposed. This algorithm significantly enhances the contrast of the ridge and the valley line, ensure the accuracy and reliability of the details of the feature extraction.

2. Usually the classic fingerprint classification algorithm needs to extract core and delta points. But in practical application, it's highly possible that delta points don't appear in the collected image, so extracting more classification features to make up for the defect will be very necessary. A new approach which combines the median line fitting with the minimum bounding ellipse method is presented to extract finger placement direction. Affine transformation is used for rotational processing after calculating the compensation of angle. In the aspect of fingerprint feature extraction, two-steps positoning method to extracting core points has been studied, which is on the basis of the orientation field consisitency coefficient and modified Poincare index algorithm. Because of the pseudo points, this thesis has proposed a sound set of methods based on human experience to help enhance the classification accuracy. Finally, a three-level fingerprint classification system based on the numbers, orientation, whorl's radius and the arch's direction of the core point, is presented. It can be used to implement the classification on the large-scale database.

3. In order to optimize the commonly fingerprint identification algorithms based on the minutiae matching, a two-level fingerprint matching algorithm is developed. First, a discrim inative sub-structure is defined for each minutia, which includes a external congruent triangle and a triangle made up of the neighbor minutiaes. Then a similarity score is employed to decide whether it is necessary to get to the global matching. If the similarity score doesn't reach a certain threshold, the potential images and candidate reference pairs which have gotten higher scores will be used in the

movement and rotation transform. Finally, we can get the global matching score. The experimental result shows that, this method greatly improve the speed and recognition rate of fingerprint identification system.

4. This thesis systemically investigates the theory and methods of AFIS. In order to verify the feasibility and practicability of the automatic fingerprint identification algorithms, it combines the object-oriented idea with database technology and developes a platform that has been constructed by using Visual Studio 2010 and MYSQL. The software has been integrated with a variety of key algorithms, and can be used to test and research on fingerprint recognition algorithm.

A large number of experimental data results show that, the algorithm modifications about the fingerprint image preprocessing can improve the mage definition effectively, and the proposed multi-level fingerprint classification approach result in high accuracy, as well as the two-level matching algorithm is effectiveness and efficiency.

Keywords: Fingerprint Identification, Large-scale Fingerprint Database, Fingerprint Image Preprocessing, Rotation Correction, Core Point Extraction, Three-Level Fingerprint Classification, Two-Level Fingerprint Identification

目录

| | |
|-------------------------------------|----|
| 摘要 | I |
| Abstract | II |
| 目录 | IV |
| 第一章 绪论 | 1 |
| 1.1 课题的研究背景及意义 | 1 |
| 1.2 课题的国内外研究现状 | 3 |
| 1.3 课题的研究内容 | 4 |
| 1.4 本文的章节安排 | 5 |
| 第二章 指纹基本特征及自动指纹识别系统介绍 | 7 |
| 2.1 引言 | 7 |
| 2.2 指纹的基本特征 | 7 |
| 2.2.1 全局特征 | 7 |
| 2.2.2 局部特征 | 9 |
| 2.2.3 细微特征 | 9 |
| 2.3 自动指纹识别系统的组成 | 10 |
| 2.3.1 指纹图像采集 | 11 |
| 2.3.2 指纹预处理 | 11 |
| 2.3.3 指纹旋转校正及分类 | 11 |
| 2.3.4 指纹特征提取 | 11 |
| 2.3.5 指纹特征匹配 | 12 |
| 2.3.6 指纹识别系统性能评估 | 12 |
| 2.4 本章小结 | 12 |
| 第三章 指纹图像的预处理及改进 | 13 |
| 3.1 预处理的必要性 | 13 |
| 3.2 指纹图像直方图均衡化 | 13 |
| 3.3 指纹图像归一化 | 15 |
| 3.4 指纹图像平滑 | 16 |
| 3.5 指纹图像分割 | 16 |
| 3.5.1 常见的图像分割方法 | 17 |
| 3.5.2 基于梯度场的图像分割方法 | 17 |
| 3.6 指纹图像方向场的计算 | 18 |
| 3.6.1 基于梯度变化的块方向场计算 | 19 |
| 3.6.2 块方向场与点方向场相结合的方法 | 20 |
| 3.7 指纹图像增强 | 22 |
| 3.7.1 Gabor 小波滤波器 | 22 |
| 3.7.2 结合 Gabor 滤波器特性的十字方向滤波方法 | 23 |
| 3.8 指纹图像二值化 | 26 |
| 3.8.1 二值化 | 26 |
| 3.8.2 二值化去噪 | 26 |
| 3.9 指纹图像骨架的细化提取 | 27 |
| 3.9.1 细化 | 27 |

| | |
|-----------------------------------|----|
| 3.9.2 细化去噪..... | 28 |
| 3.10 本章小结..... | 28 |
| 第四章 指纹图像多级分类..... | 29 |
| 4.1 指纹分类基础..... | 29 |
| 4.2 指纹分类方法性能评估..... | 29 |
| 4.3 常见的指纹分类方法..... | 30 |
| 4.4 指纹旋转校正..... | 31 |
| 4.4.1 校正指纹图像的必要性..... | 31 |
| 4.4.2 文献中的旋转角度确定方法..... | 32 |
| 4.4.3 本文改进的方法..... | 33 |
| 4.4.4 依据旋转角度旋转图像..... | 37 |
| 4.4.5 实验结果及分析..... | 38 |
| 4.5 奇异点提取..... | 40 |
| 4.5.1 传统的 Poincare index 方法..... | 40 |
| 4.5.2 本文采用的提取方法..... | 41 |
| 4.5.2.1 确定中心点所在的区域..... | 41 |
| 4.5.2.2 改进的 Poincare 方法提取中心点..... | 42 |
| 4.5.2.3 计算中心点角度..... | 43 |
| 4.5.2.4 去伪中心点..... | 44 |
| 4.5.3 实验结果及分析..... | 45 |
| 4.6 本文的分类方法..... | 47 |
| 4.6.1 一级分类方法..... | 47 |
| 4.6.2 二级分类方法..... | 48 |
| 4.6.3 三级分类方法..... | 49 |
| 4.6.3.1 针对斗型指纹三级分类..... | 49 |
| 4.6.3.2 针对簸箕型指纹三级分类..... | 50 |
| 4.7 实验结果及分析..... | 51 |
| 4.8 本章小结..... | 52 |
| 第五章 基于点模式的指纹特征匹配..... | 53 |
| 5.1 指纹特征匹配概述..... | 53 |
| 5.2 细节点提取..... | 53 |
| 5.3 细节点去伪..... | 54 |
| 5.4 特征向量的表示..... | 55 |
| 5.5 初匹配..... | 57 |
| 5.6 全局匹配..... | 58 |
| 5.7 实验结果及分析..... | 60 |
| 5.7.1 时间复杂度分析..... | 60 |
| 5.7.2 实验结果分析..... | 60 |
| 5.8 本章小结..... | 61 |
| 第六章 自动指纹识别系统软件的设计..... | 63 |
| 6.1 软件平台搭建..... | 63 |
| 6.2 自动指纹识别系统软件整体设计..... | 63 |
| 6.2.1 基于界面结构划分..... | 63 |
| 6.2.2 基于面向对象的类划分..... | 64 |

| | |
|----------------------------|----|
| 6.3 系统平台各功能模块详述..... | 65 |
| 6.3.1 图像读取保存模块..... | 65 |
| 6.3.2 指纹图像处理算法研究模块..... | 65 |
| 6.3.3 指纹图像处理算法应用模块..... | 65 |
| 6.3.4 运行结果显示模块..... | 67 |
| 6.4 本章小结..... | 68 |
| 第七章 总结与展望..... | 71 |
| 7.1 工作总结..... | 71 |
| 7.2 展望..... | 72 |
| 参考文献..... | 73 |
| 致谢..... | 77 |
| 作者在攻读硕士学位期间发表的论文和获奖情况..... | 79 |

第一章 绪论

1.1 课题的研究背景及意义

科学技术的快速发展给人们的生活带来了深刻的影响,改变了传统的生活方式,人们之间的交流越来越方便快捷。然而,在享受着高新科技带来便利的同时,随之而来的是日益加剧的信息安全问题。传统的以简单密码为特征的身份认证方法越来越难以满足人们对高安全性的要求,像反恐、机场、政府机构、金融、军事、智能家居等行业,其身份认证方式往往是通过标志物(如钥匙、证件、磁卡)或知识(如口令、密码)来达到鉴别的目的,这些方式都是通过以物认人来间接进行的,都存在易丢失、易被盗、易被伪造等缺陷。因此,找到一种更加便捷、更加安全可靠的身份验证方式成为了一个亟待解决的问题。

生物识别技术(Biometric Identification Technology)是依据个人相对稳定的生理特征和行为特征来完成个人身份认证的一种技术,能够克服传统身份认证方式的许多缺点。在多种用来区别身份的生物特征中,生理特征是与生俱来的,目前主要包括:指纹、掌纹、虹膜、视网膜、人脸、DNA等^{[1][2]}。相反,行为特征是后天形成的,具有一定的可变性,主要是声音、步态、签名等。两者比较来说,生物特征具有更实际的应用价值。图1-1展示了常见的生物特征。

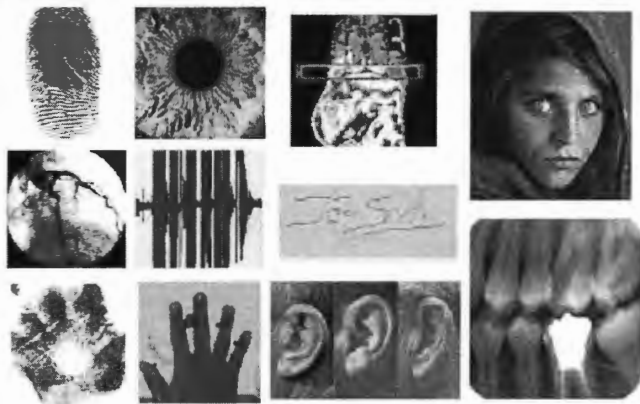


图1-1 常见的生物特征

图1-1所列举的生物特征并不是都适用于生物识别的,要作为身份鉴别的依据,还应该满足以下几个条件^[2]:

- (1) 普遍性(Universality): 是否每个人都具有这种生物特征;
- (2) 唯一性(Uniqueness): 每个人具有的这种生物特征是否是独一无二的;
- (3) 持久性(Permanence): 是否这种生物特征不会随着年龄发生很大的变化;
- (4) 可采集性(Collectability): 是否这种生物特征容易被采集到;
- (5) 可接受程度(User Acceptance): 使用者是否愿意接受这种身份认证方式;
- (6) 防欺骗性(Resistance to Circumvention): 是否这种身份认证方式容易受环境影响,是否可以防止欺骗。

在各种生物识别技术中,指纹和掌纹识别应用最为普遍,具有采集方便,较易被大众所接受的优点;虹膜识别具有高准确性和防伪性,但是它没有进行任何

的测试，只是用统计学原理进行过小范围的试验；人脸识别和语音识别在未来智能设备中的应用前景广阔，目前仍处于发展阶段；DNA 识别是一种成熟的、精确的生物识别技术，但因基因检测技术较为复杂，还无法走向民用；签名鉴定分为在线鉴定和离线鉴定，离线鉴定比较容易伪造，识别难度大，在线签名相对来说具有更高的安全性，但因为人类书写动力定型并非固定不变，所以签名鉴定实际上是一种准确性很低的生物识别技术。

目前还无法确定地说哪一种方法占据绝对优势，通过表 1-1 分别对几种常用的生物识别方法做性能的分析 and 比较。

表1-1 人体中常见的生物特征性能比较

| 生物特征 | 普遍性 | 唯一性 | 持久性 | 可采集性 | 可接受程度 | 防欺骗性 |
|------|-----|-----|-----|------|-------|------|
| 指纹 | 中 | 高 | 高 | 中 | 中 | 高 |
| 掌纹 | 中 | 中 | 中 | 高 | 中 | 中 |
| 虹膜 | 高 | 高 | 高 | 中 | 低 | 高 |
| 人脸 | 高 | 中 | 低 | 高 | 高 | 低 |
| DNA | 高 | 高 | 高 | 低 | 低 | 高 |
| 声音 | 中 | 低 | 低 | 中 | 高 | 低 |
| 签名 | 低 | 低 | 低 | 高 | 高 | 低 |

图 1-2 是采用 Zephyr 分析法^[3]对生物识别技术进行更加直观的统计和比较。图中不同的标号分别表示干扰用户程度、唯一性、费用和易用性。标号相对中心的距离代表该方法所达到的水平，性能越高，对应的标号就离中心越远。

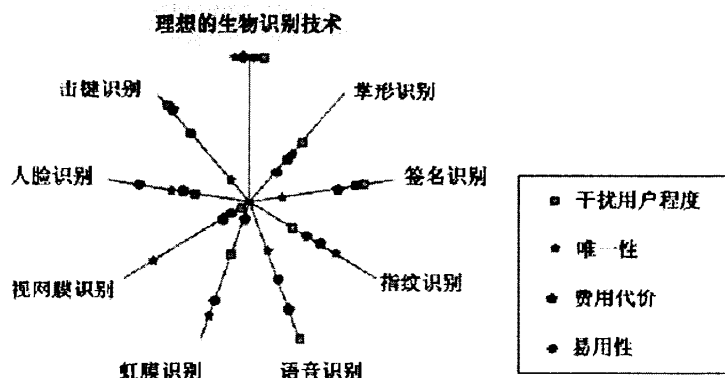


图1-2 生物识别技术的 Zephyr 分析^[3]

以上一系列的分析说明，指纹识别技术成熟稳定、成本较低，优势明显，是一种比较理想的生物特征识别技术。目前，还未发现任何两枚手指具有相同的指纹纹路，即使 DNA 具有很高的相似度，其指纹也会不同^[4]。另外，指纹的采集也十分地快速和便捷，对人类没有任何伤害。从应用开发的角度，指纹的存储量也非常小，可以方便地移植到嵌入式应用中。所以，自动指纹识别系统的研究具有显著的学术价值和应用前景。

在过去的一百多年里，指纹识别因其多方面的优点已被世界各地广泛应用。早在我国古代时期，人们就已经用指纹作为一种签押的方式。现今，指纹识别技术更加成熟，不断应用在各个领域：除了刑事侦查之外，在民用领域也发挥着重要的作用，如智能家居的指纹门禁系统、驾驶员指纹管理系统、公司考勤系统、

银行指纹保险箱、笔记本电脑、手机、金融交易支付等生活的各个方面^{[3][5]}。目前比较热门的应用还包括：美国苹果公司生产了应用指纹识别解锁的智能手机、荷兰连锁超市启用指纹消费功能、国内的华为公司与支付宝联合实现手机指纹识别在线支付等。指纹识别与人们的生活联系越来越紧密，给日常生活带来了极大的便利和安全性。图 1-3 展示了互联网时代指纹识别技术新的应用场景。



图1-3 互联网时代指纹识别技术的应用场景

1.2 课题的国内外研究现状

目前，最早关于指纹鉴定身份的记录可以追溯到公元前 221-206 年，早在中国古代的秦朝，人们就利用指纹纹路具有差异性的特点侦查犯罪^[6]。尽管古人很早就意识到指纹的唯一性，但因为缺乏专门的指纹识别技术，一直无法有更深入的科学的研究。直到 1684 年，来自英国的学者 Nehemiah Grew 首次公开发表有关指纹的专业论文，主要对指纹的纹线和毛孔结构进行了全面的阐述，这标志着科学化、现代化的指纹研究的开始^{[7][8]}。到了 1788 年，德国解剖学家 Mayer 从解剖学角度详尽地刻画了指纹的模式形态，第一次指出了指纹的唯一性^{[9][10]}。1880 年，Henry Fauld 首次提出了指纹特征脊线形态的重要特征——一致性和永久性^[11]。1984 年，英国人 Edward Henry 在《指纹的分类和应用》一书中阐述了著名的 Henry 五类分类体系方法，该分类系统很快被推广到全世界，一直沿用至今^[7]。从此，指纹分类研究进入到一个新的阶段，得到了更多的关注。进入 20 世纪以后，指纹识别成为了西方国家法定的身份鉴定方式。1924 年，美国联邦调查局 FBI (Federal Bureau of Investigation) 首次建立了数量大约 81 万的指纹样本库，并逐年增加。随着指纹数据库容量的扩大，如此海量的指纹数据仅仅依靠人工识别已很难满足实际的应用需求。计算机的不断发展给指纹识别带来了新的变革，从 20 世纪 70 年代开始，发达国家相继引入计算机技术完成对指纹的识别和处理，纷纷开始了对自动指纹识别系统 (Automated Fingerprint Identification System, AFIS) 的研究。例如，美国于 1975 年成功推出第一个商业化系统 Printrak250，日本的 NEC 公司在 1982 年就向警方提供了自动指纹识别系统，大大提高了司法部门的工作效率，并推动了自动指纹识别系统往民用方向的发展^[12]。

指纹识别的研究在国外开展的较早，并且应用更加广泛。目前，从事自动指纹识别研究的公司、科研机构和学校的数目正逐年增加。这其中，美国国家标准局视觉图像处理研究组 (Visual Image Processing Group)、IBM 沃特森研究中心 (IBM TJ Watson Research Center)、密歇根州立大学 (Michigan State University) 的模式识别与图像处理实验室 (Pattern Recognition and Image Processing Lab) 等机构都在指纹处理的各个领域有所成就。国际上也设立了有关指纹识别的多项竞赛，其中 FVC (International Competition Fingerprint Verification Algorithm) 是全球生物特征识别技术行业里最高级别的竞技大赛，目的就是全面了解、评价

当前指纹识别技术的发展水平,给企业和研究机构建立一个评价指纹识别算法的通用标准^[13]。

我国的指纹识别研究开始于 80 年代,清华大学、中科院自动化所都一直致力于生物识别技术。目前,中科院自动化所是我国最权威的指纹识别科研机构,曾经在 FVC2004 国际指纹竞赛中取得优异成绩。以指纹识别为首的生物特征识别技术也已经被列为国家重点发展的高新技术之一。国务院在 2006 年出台的《国家中长期科学和技术发展规划纲要》(2006-2020)^[14]中指出要“重点研究生物特征识别、智能交通等相关领域的系统技术”。近年来,随着指纹采集设备的成本越来越低,技术越来越先进,国内涌现了多个生物识别技术公司和网站,并开发出了大量应用于民用的考勤机、智能门锁等指纹识别产品。总体来说,虽然国内自动指纹识别系统的起步稍晚,但取得的成就是举世瞩目的,在互联网时代来临的关键时刻,大批有关指纹识别的新型应用产品正逐步走进人们的生活。

尽管指纹识别的研究已相对成熟,也应用于不少领域,但是在准确率和速度等方面还存在着许多问题,因此还不能满足很多实际应用的要求。面对互联网潮流下的大数据热门问题,传统的指纹识别算法也面临着巨大的挑战。根据应用场景的不同,主要亟待解决两方面的问题:一是针对传统应用,如智能家居指纹锁,如何对低价格采集设备采集到的指纹图像进行又快又准的特征匹配是需要不断完善和提高的地方;二是实现海量指纹图像检索,在指纹识别之前,先要对指纹图像进行分类,能否正确建立完善的索引机制是一个具有挑战性的研究课题。

1.3 课题的研究内容

根据上文所述,本课题在研究指纹识别通用算法的同时,考虑到大规模数据库的应用背景,重点分析研究了指纹图像预处理算法、指纹多级分类及相关算法、指纹两级匹配的快速方法。具体内容如下:

1. 指纹预处理算法及改进

预处理是指纹采集后的第一步,是后续算法的基础。在指纹图像预处理阶段,主要进行了包括图像均衡化、归一化、平滑、图像分割、方向场计算、图像增强、二值化及去噪、细化及去噪等一系列工作,并重点对直方图均衡化、方向场计算和图像增强进行了改进,提高了算法的鲁棒性。实验结果证明本文的改进起到了非常重要的作用。

2. 旋转校正和特征提取

传统的指纹分类方法通常需要提取三角点和中心点,但并不是所有的指纹采集仪的采集面积都足够覆盖三角点,所以本文提出了一种将发生旋转的指纹图像进行旋转校正的方法,以获得更多其他的全局特征,弥补提取不到三角点时无法分类的缺点。同时,在阅读了大量文献后,建立了一整套中心点提取和去伪的算法,能够快速准确地提取中心点,保证指纹分类的顺利进行。

3. 指纹分类

在经典 Henry 分类模型基础上,研究了一种基于指纹中心点及邻近全局特征的多级指纹分类方法。首先根据中心点的类型、数目、方向等设计了一二级的排斥性分类,然后对出现概率较大的斗型和箕型提出了三级连续性分类方法,可以将指纹分成近百类。实验结果表明分类过程准确、快速。

4. 指纹匹配

本文主要采用改进的基于点模式的匹配模型进行特征匹配,分为初匹配和全

局匹配两级。针对点模式匹配的关键问题：参考点对提取，设计了将细节点方向场分布和相邻细节点三角形方法相结合的初匹配方案，帮助全局匹配节约了大量的时间。并且，当初匹配阈值高于一定阈值时，直接就可以退出匹配模块。实验结果表明，本文改进的方法在识别速度和正确率方面有了很大的提高。

5. 自动指纹识别系统的开发

为了便于研究和验证自动指纹识别系统的相关算法，开发了一套基于 VS2010 和 MYSQL 数据库的软件平台。根据功能划分，该软件主要包括四个模块：图像读取保存模块、指纹图像算法研究模块、指纹图像算法应用模块和运行结果显示模块。

1.4 本文的章节安排

本文共分为七章，各章节内容安排如下：

第一章是绪论。本章就课题的研究背景及研究意义进行了概述，分析了指纹分类和识别相关算法的国内外研究现状，概要介绍了课题的主要研究内容。

第二章是指纹基本特征及自动指纹识别系统介绍。本章介绍了指纹的基本特征，包括全局特征、局部特征和细微特征，然后给出了自动指纹识别系统的总体设计和各模块内容的说明。

第三章是指纹图像的预处理算法及改进。本章阐述了预处理算法的思想和实现，并对重要的算法进行改进。

第四章是指纹图像多级分类。本章分析了大规模指纹数据库中指纹分类的重要性，并提出了基于指纹旋转校正、中心点提取的多级指纹分类算法。

第五章是基于点模式的指纹特征匹配。本章简要分析了文献中基于点模式的特征匹配方法，着重提出了一种改进的两级匹配算法。

第六章是自动指纹识别系统软件的设计。本章给出了基于 VS2010 和 MYSQL 数据库开发的指纹识别系统软件，该软件采用了面向对象思想和数据库设计方法，可用于指纹算法的研究和应用。

第七章是结束语。本章总结了本文的主要工作，陈述了作者在本课题研究工作中的心得体会，对指纹分类和匹配相关算法尚待完善的地方及未来的发展方向作出了展望。

此页不缺内容

第二章 指纹基本特征及自动指纹识别系统介绍

2.1 引言

自动指纹识别技术是指以指纹作为生物特征的生物特征识别技术。人类使用指纹识别进行个人身份鉴定已经有很长的历史，早期是由人工肉眼对指纹卡进行辨识。随着科技的发展，人工识别已远远满足不了现代社会生活的需求。相反，基于计算机的自动识别指纹系统有着许多显著的优点，如计算能力和存储容量更为优秀，制造成本逐年降低，采集传感器技术不断发展等^[15]。

本章在介绍自动指纹识别系统之前，首先对人类指纹的基本特征进行简单的阐述，然后针对自动指纹识别系统的总体框架和各部分组成进行说明。

2.2 指纹的基本特征

指纹是指手指正面皮肤凹凸不平的纹路，纹路中隆起部分是手指真皮向表皮乳突形成的皱痕，又称指纹脊线（ridge），指纹脊线之间的凹陷部分，称为指纹谷线（furrow）^[16]，如图 2-1。不同的指纹，其纹线数目、走向和相对位置一定是存在差异的，并且这些特征随着人的年龄增长变化不大。指纹的图像识别算法归根结底就是为待识别指纹找到具有相同或近似图像特征的样本指纹，所以本章要重点介绍指纹图像的基本特征。



图2-1 指纹的脊线和谷线示意图

指纹的脊线是其最直观的结构特征，根据描述层次的不同，可以将脊线特征分为三类：全局特征、局部特征和细微特征^[15]。一般在指纹识别系统中要综合至少两种类型的特征，因为任意两枚指纹的全局特征或者局部特征可以很相近，但要满足全局特征和局部特征同时相同是几乎不可能的。

2.2.1 全局特征

指纹的全局特征结构包括人眼直接就可以观察到的特征，比如指纹纹型、模式区、中心区、三角区以及纹线数目等^[17]。

其中，纹路纹型（简称纹型）是最基本的分类方式。目前比较常见的分类方法是依据指纹的中心模式区和三角模式区划分的，纹形则是指纹的纹型的子类，即纹型和纹形是类与种的关系^[17]。一般根据指纹形状的不同将指纹分为 3 大类型（弓型纹、箕型纹和螺旋型纹），以及 9 种形态（弧形、帐形、左簸箕形、右

簸箕形、环形、螺形、囊形、双箕形、杂形)，这种分类方法也被称为两级分类法^{[18][19]}。



图2-2 纹型示意图

根据图 2-2 所示，弓型纹纹线从手指一侧流入，在手指中部向上隆起，然后从另一侧流出，纹线方向变化较小。箕型纹纹线从手指一侧流入，在手指中央弯曲并折回，然后从同一侧流出。螺旋型纹的中心区域纹线是环形或者螺旋形的曲线，由内向外扩散并与上下纹线汇合而成。表 2-1 为相关文献统计得出的纹型分布规律，从中可以得出结论：绝大部分的指纹都是属于箕型纹和螺旋型纹的^{[18][19][20]}。

表2-1 自然界中指纹纹型分布概率

| 纹型 | 弓型纹 | 箕型纹 | 螺旋型 |
|------|-----|-----|-----|
| 分布概率 | 5% | 60% | 35% |

为了更进一步的细化分类，需要介绍几个相关的概念：模式区 (Pattern Area)、奇异点 (Singular Point)、中心点 (Core Point)、三角点 (Delta) 和纹线数目 (Ridge Count) ^[21]。

1. 模式区 (Pattern Area)：模式区是指包括了指纹全局特征的前景区域，一般指纹分类特征都是从模式区提取的。一些指纹识别算法为了减少需要比对的特征向量数目，加快识别速度，仅仅使用待测指纹模式区的信息进行特征匹配；另外一些算法则使用整个指纹的所有特征以提高识别正确率。模式区的特征如图 2-3。

2. 奇异点 (Singular Point)：奇异点是指在模式区中指纹纹线方向变化较大的特征点，主要包括中心点和三角点。

3. 中心点 (Core Point)：中心点位于纹线渐近中心，常常被用作指纹分类的重要特征。中心点位置如图 2-3。

4. 三角点 (Delta)：三角点是以中心点为开始点遇到的分叉点或者断点、或两条纹路的汇聚孤立处和转折处。三角点常常作为指纹图像纹路的计数与追踪的开始之处。三角点位置如图 2-3。

5. 纹线数目 (Ridge Count)：纹线数目是指指纹图像模式区内指纹纹线的数量。可以通过连接中心点和三角点，统计这条连线与指纹脊线相交的个数获得纹线数目。



图2-3 指纹全局特征

2.2.2 局部特征

通常，脊线上的一些点能够有效地表示指纹图像中较少改变的、具有高可信度和区分度的局部特征信息，这种脊线上的局部特征被称为细节点特征 (Minutiae) [15]。图 2-4 给出了常见的细节点类型。

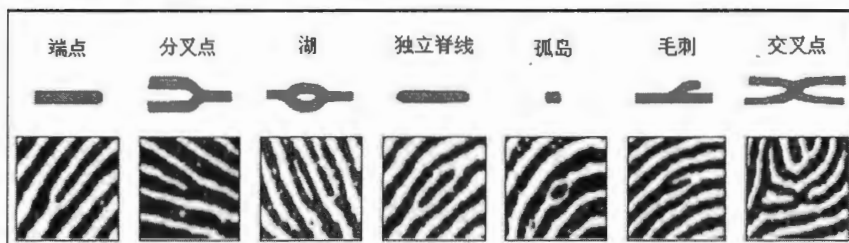


图2-4 常见局部细节点类型

- 端点：一条脊线的终结点；
- 分叉点：一条指纹纹线在此点分成两条或更多条的指纹纹路；
- 湖：存在两条或者两条以上的连接两个分叉点的纹线；
- 独立脊线：一条较短但不足以成为一点的纹线；
- 孤岛：一条比较短的纹线，可以看做是一点；
- 毛刺：一条指纹纹线在此处分出一条较短但不至于成为脊线的纹线；
- 交叉点：四条纹线汇聚在此点。

在以上所述的细节特征中，有些特征是提取不到或者不能作为匹配特征的，而且端点和分叉点出现的概率是所有局部特征的 80% 以上，还具有易提取、对噪声不敏感等优点。所以本文主要关注这两种局部特征。

2.2.3 细微特征

针对采用高分辨率传感器（当分辨率达到 1000dpi 以上）采集的指纹图像，还可以继续提取脊线上更加细微的特征[8]。在实时性要求不高、安全性要求较高的应用场合，细微特征可以有效地提高自动指纹识别系统的识别准确率，但一般的采集设备分辨率只能达到 500dpi，所以自动指纹识别系统中通常并不提取细微特征。常见的细微特征是汗腺孔特征，每个人的汗腺孔排列形状都是独一无二且永恒不变的[22]。汗腺孔具体细节如图 2-5 所示。



图2-5 汗腺孔特征示例图

2.3 自动指纹识别系统的组成

自动指纹识别系统 AFIS (Automatic Fingerprint Identification System) 本质上是一个模式识别系统，并且是集光电技术、图像处理技术、数据库技术、模式识别技术、人工智能技术等多种技术于一体的综合性系统^[23]。其工作模式可以分为验证模式 (Verification) 和识别模式 (Identification) 两类，本文重点关注识别模式，也就是一对多识别 (one-to-many matching) 方式。其核心思想就是搜索样本数据库中的所有指纹，与录入指纹逐一进行对比，直到找出相匹配的指纹图像或者给出没有匹配指纹的结论。

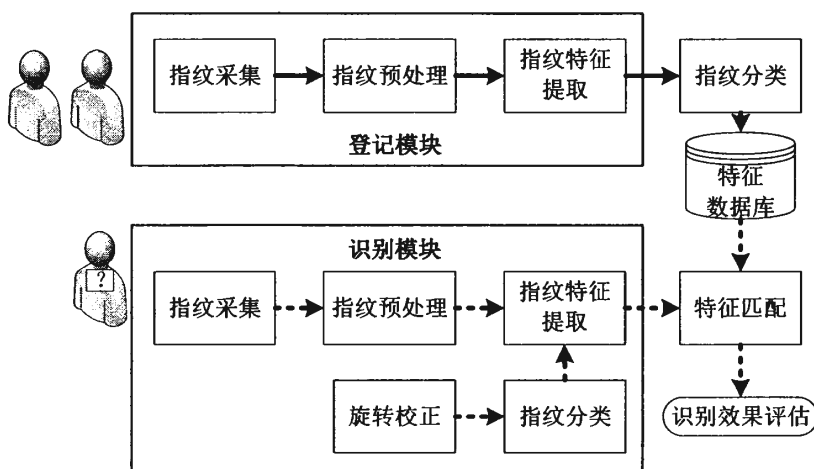


图2-6 典型的自动指纹识别过程

基于识别模式的自动指纹识别系统从功能上讲，由指纹登记和指纹识别两个模块组成；从结构上讲，又可细分为六个部分：指纹采集设备、样本存储数据库、图像预处理、特征提取、特征匹配和系统性能的效果评估。根据实际需求，还可以包括其他部分，如指纹旋转校正、指纹分类等，典型的自动指纹识别系统流程如图 2-6。指纹登记模块主要负责将指纹特征信息登记到样本数据库中，在登记过程中，指纹样本通过设备采集、图像预处理，进而经特征提取获得特征向量信息，根据全局特征对指纹分类，最终得到指纹样本数据库。根据不同的需求，数据库被保存在 PC 端或者 FLASH 芯片中。指纹识别模块主要负责判断待识别指纹与样本数据库中的指纹是否匹配。在该模块中，采集设备获取待识别的指纹图像，然后进行预处理，若指纹图像发生了旋转，还需要对偏转的指纹对象进行校正，然后经特征提取模块提取待识指纹的特征信息，经分类模块分类后，利用特征匹配算法将这些信息与指纹数据库中的样本特征信息进行逐一比对，给出是

否匹配成功的结论。下面对自动指纹识别系统的六个组成部分进行简单的介绍。

2.3.1 指纹图像采集

在自动指纹识别系统中,第一项工作就是使用指纹采集仪采集活体指纹图像。早期的指纹图像采集是通过扫描纸张上的指纹图来完成的,这种方式被称为离线方式;随着传感器技术、半导体技术和数字图像处理技术的快速发展,现在基本上都是通过指纹传感器采集指纹,也被称为在线方式^[19]。

在指纹采集仪中,最重要的组成部件就是传感器。通常,依据采集原理的不同可以将常用的指纹传感器分为三种:光学传感器、固态传感器和超声波传感器^{[15][16][17][18][19]}。光学式采集仪利用全反射原理,根据指纹脊线、谷线与玻璃接触面的距离不同采集指纹图像,其产生较早,采集面积大,可靠性高,但对温度等环境因素的适应力差,易发生光学畸变;电容式指纹仪是典型的固态式采集仪,采集时手指直接接触硅面,通过微小传感器将一个个像素点转换成电信号,优点是价格低廉,采集的图像质量较好,但采集面积不大,容易损坏;超声波式采集仪是直接向手指发送并捕捉返回的超声波信号,根据深度的不同有效区分脊线和谷线,抗噪声效果好,支持非接触式采集,但价格昂贵,采集时间较长,不适用于民用设备。

本文采用经济实用的电容式指纹传感器进行指纹图像的采集。

2.3.2 指纹预处理

一般通过指纹采集仪获得的指纹图像是灰度图像,受采集环境、人为因素等影响,往往无法直接提取到有效的指纹特征,还需要进行预处理操作。指纹图像预处理是利用数字图像处理技术消除指纹纹线中的噪声干扰,分割出有用的指纹前景图像,计算指纹纹线的方向图,把整幅图像变成纹理清楚的纹线图像。预处理操作是整个自动指纹识别系统中最为重要的一步,直接影响到最终的识别性能。

2.3.3 指纹旋转校正及分类

当指纹识别应用于大规模数据库时,如果没有指纹分类模块,待识别指纹图像就需要与样本数据库上所有的指纹进行一一比对,时间复杂度可想而知。为了减小图像匹配的工作量,本文对采集到的指纹图像进行多级分类,这样在搜索数据库时只需要比对相对的一个子集就可以了,满足指纹匹配实时性的要求。

指纹分类的方法有很多,最广泛的就是利用指纹的奇异点的方向、位置、相对距离等特征,基于决策树模型进行分类。但三角点受采集仪采集面积的限制,并不是每一幅图像中都存在三角点,为了弥补这个缺点,进一步提高分类精度,还需要确定中心区的绝对方向。通常,我们采集到的图像有时会发生旋转,导致无法计算纹线走向的绝对角度,所以,在分类之前先要对旋转的指纹图像进行校正,保证对称轴没有倾斜,方便提取特征信息。

2.3.4 指纹特征提取

指纹特征提取操作是从经过图像预处理后的指纹图像中提取用于分类的全局特征和用于匹配的局部特征的过程。其中,全局特征直接从旋转校正后的指纹图像方向图中就可以提取,考虑到全局特征提取的准确率直接关系到指纹分类的性能优劣,所以还需要对全局特征里的奇异点集进行有效的去伪操作,减小错误

的奇异点干扰指纹分类的可能性。而局部特征点一般用于特征匹配，其提取过程涉及到灰度值的计算，为了降低算法复杂度，通常从预处理后的细化指纹图像中提取细节特征。

两种特征的提取并不是相互独立的，选取特征点的方法有很多，比较常见的做法是以指纹的中心点为圆心，由近及远地选取一定数目的细节特征点用于特征匹配。一般，在民用指纹识别系统的应用中，一幅指纹图像可以选取 1~3 个中心点，20~50 个细节特征点。

2.3.5 指纹特征匹配

指纹匹配是将输入待识指纹的特征向量与样本库中所存储的指纹样本特征向量进行比对，找出最相似的指纹作为识别的输出结果^[21]，它是指纹识别技术的核心内容，匹配算法的准确率和算法复杂度直接决定了自动识别系统的性能好坏。

2.3.6 指纹识别系统性能评估

在指纹识别系统中，比对两幅指纹图像的特征向量后，匹配模块输出的结果往往是一个相似度值。通常，为指纹识别系统设定一个合适的阈值 ε ，当相似度小于 ε 时，认为两幅图像不是出自同一枚手指，当相似度大于等于 ε 时，则认定两幅指纹图像匹配成功。在实际应用中，计算机处理指纹时只是使用了提取出的一些局部特征信息，其结果并不能保证 100% 精确。

所以，指纹识别系统采用识别率作为衡量性能的指标，识别率又分为误识率（False Accept Rate, FAR）和拒识率（False Reject Rate, FRR）。具体来说，误识率（FAR）是指不属于同一枚手指的指纹被错误匹配的次數与所有比对次數的比值，它实际上是系统的安全度测量指标；拒识率（FRR）指属于同一枚手指的指纹被判断为不属于同一枚手指的次數与总对比次數的比值，它是易用性的衡量标准^[21]。实际上，FAR 与 FRR 是成反比例的^[19]，在实际应用中，要根据不同的用途调整两个值。若产品的定位是高可靠性、低实用性，则需要设计过程中降低 FAR；若产品不需要很高的安全性，但是要方便使用，则以较低的 FRR 为前提设计算法。综上，可以根据不同领域的应用，在两指标之间取折中。

2.4 本章小结

本章首先介绍了指纹的全局特征、局部特征和细微特征，然后简要地描述了自动指纹识别系统中各个阶段的处理和基本原理：包括指纹图像采集、指纹图像预处理、指纹特征提取、指纹分类、指纹匹配和指纹识别系统评估标准，这些是后续内容的理论基础，具体实现将在后面的章节中详细介绍。

第三章 指纹图像的预处理及改进

3.1 预处理的必要性

无论是对指纹进行分类还是匹配识别, 指纹图像的预处理操作都是必不可少的, 其直接关系到整个系统的处理效果。通常, 我们希望得到一幅纹路清楚、灰度对比度明显的指纹图像, 因为只有在这样的指纹图像中才能将指纹特征点的位置和类型准确确定下来, 但通过指纹采集传感器获得的指纹图像一般都是被各种类型噪声所污染的灰度图像。这些噪声主要包括脊线的断裂和粘连, 破坏了指纹真实的脊线结构, 使得在进行特征提取时, 无法提取正确的特征点, 而且产生大量错误结果, 从而降低了整个识别系统的性能。

预处理的目的是在保留原有图像中重要信息的情况下, 尽可能地去掉原始指纹图像中的冗余和噪声信息, 增强图像的细节特征, 为后续的处理做好准备。通常指纹图像预处理过程包括直方图均衡化、图像归一化、图像平滑、图像分割、方向场计算、图像增强、二值化及去噪、细化及去噪等步骤, 工作过程如图 3-1 所示。

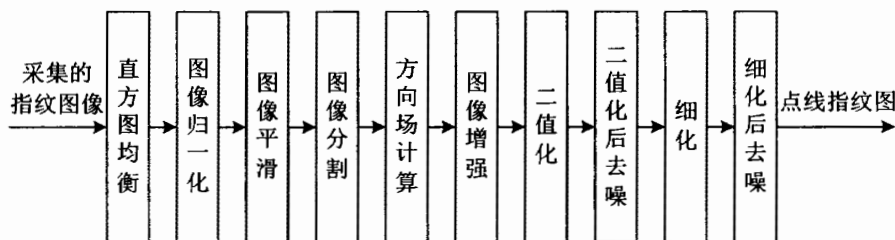


图3-1 指纹预处理过程

3.2 指纹图像直方图均衡化

我们的手会接触各种表面, 手指往往会出现潮湿、油腻、有断痕的情况, 这容易造成下面几种问题: 1) 遗漏真实信息; 2) 带来大量伪特征; 3) 真实特征点的方向计算错误、位置偏移。究其原因, 是图像的对比度过低, 纹线与脊线无法清晰地分离开。

为了提高图像的灰度对比度, 传统的方法是采用空间域法, 进行直方图均衡化, 其算法简洁, 处理速度快, 缺点就是因为考虑的是全局的灰度值, 图像中局部细节往往得不到有效的增强, 那些在灰度直方图中占低百分比的局部细节, 极易在均衡化操作中被邻域的灰度级融合, 反而降低了图像的对比度^[24], 图 3-2(a) 为指纹采集的原图, 图 3-2(b) 为传统直方图均衡化的效果图。文献[25]中使用了分块直方图均衡化的方法, 虽然考虑到了局部的细节, 但是由于进行了分块从而导致整体性并不好, 使得图像经过处理后容易发生过增强现象, 需要进一步考虑 8 邻域联通性, 增加了计算复杂度, 效果如图 3-2(c)。文献[26]采用灰度拉伸方法, 即利用指数函数或对数函数将图像灰度级压缩, 进而达到提高对比度的目的, 效果如图 3-2(d), 可以看出其对比度有了显著的提高, 但代价是较细的指纹或者

断裂的指纹在后续图像增强处理中无法得到修复，并且容易带来伪特征。



图3-2 常见的增强对比度方法结果图

本文对灰度直方图方法进行了改进，在统计图像中各灰度值的概率时并不统计所有的灰度值。首先需要计算指纹图像的灰度平均值，记输入的灰度指纹图像 I 为 $M \times N$ 的矩阵， $I(i, j)$ 代表第 i 行第 j 列的像素灰度，指纹图像的灰度均值计算公式如下：

$$M(I) = \frac{1}{MN} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} I(i, j) \quad (3.1)$$

然后，根据反复实验的结果，利用灰度平均值 $M(I)$ 设置阈值 T ：

$$T = \begin{cases} M(I) + 30 & 0 \leq M(I) \leq 170 \\ 190 & 170 < M(I) < 190 \\ M(I) + 5 & 170 < M(I) < 190 \\ M(I) & else \end{cases} \quad (3.2)$$

接着，定义数字图像 I 中灰度值小于 T 的概率：

$$P(k) = n_k / n_T, \quad 0 \leq k \leq T \leq L-1 \quad (3.3)$$

其中， n_k 是灰度值为 k 的像素数目； n_T 为满足阈值条件的像素总数 ($n_T \leq MN$)； L 是图像中可能的灰度级的数量（本文中是 8bit 图像，即 256）。

最后，根据公式 (3.4) 依次更新每个像素点新的灰度值：

$$s_k = (L-1) \sum_{i=0}^k p_i, \quad 0 \leq k \leq T \leq L-1 \quad (3.4)$$

这样，将输入图像中灰度级为 k 的像素点 $I_k(i, j)$ 映射到输出图像灰度级为 s_k 的像素点。实验结果如图 3-3，(a) 为采集的指纹原图，(b) 为改进的直方图均

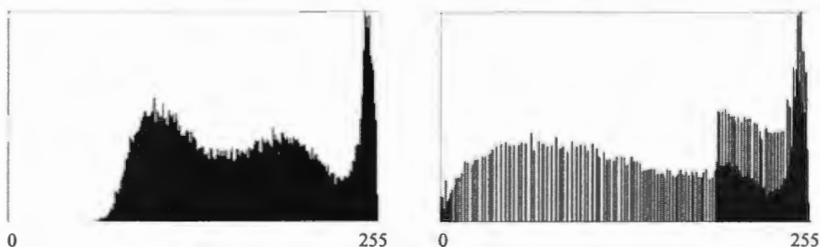
衡化处理结果图，在对比度提高的情况下仍保持原图像的灰度级范围。



(a) 是原始图像

(b) 改进的均衡化方法

图3-3 改进的均衡化方法处理结果图



(a) 原始图像灰度直方图

(b)处理后的灰度直方图

图3-4 指纹图像处理前后的灰度直方图对比

通过处理前后的灰度直方图 3-4 可以看出，原本密集的像素灰度分布被均匀地扩展到整个 0~255 范围内，直方图更加平坦。在指纹图像中，原本灰度值较高的背景区保持原有的亮度，同时前景区的脊线得到了灰度增强，整幅图像的对比度明显提高，细节特征得到了突出。由此，可以认为这种改进的直方图均衡方法能够有效地抑制指纹粘连带来的伪特征，为后续求取正确的方向图做好了准备。

3.3 指纹图像归一化

通常采集到的指纹图像灰度范围参差不齐，如果可以将不同指纹图像的灰度级都映射到同一个固定的级别上，就能为后续算法提供一个较为统一的图像规格^[21]，方便设置与灰度相关的阈值。因此，本文还进行了图像归一化处理。

根据文献[27]，指纹图像的灰度均值和方差公式定义如下：

$$M(I) = \frac{1}{MN} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} I(i, j) \quad (3.5)$$

$$\sigma^2(I) = \frac{1}{MN} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} [I(i, j) - M(I)]^2 \quad (3.6)$$

用 $G(i, j)$ 表示归一化后位置 (i, j) 处的灰度值，归一化计算如下：

$$G(i, j) = \begin{cases} M_0 + \sqrt{\frac{\sigma_0^2 (I(i, j) - M)^2}{\sigma^2}} & I(i, j) > M \\ M_0 - \sqrt{\frac{\sigma_0^2 (I(i, j) - M)^2}{\sigma^2}} & I(i, j) \leq M \end{cases} \quad (3.7)$$

式中, M_0 和 σ_0^2 是预先设定的灰度均值和方差 (考虑分辨率, 灰度均值等因素令 M_0 为 150, σ_0^2 为 70), M 和 σ^2 为原始图像的灰度均值和方差。实验中使用的指纹图像大小为 256×288 , 灰度级为 256, 图像归一化效果如图 3-5。



图3-5 图像归一化后效果图

实验结果表明, 归一化能够有效降低图像中脊线、谷线方向的灰度变化程度, 减小了动态变化范围, 但并不能改善脊线和谷线结构的清晰对比度。

3.4 指纹图像平滑

在采集指纹图像的过程中, 传感器常会把周围环境中的灰尘和污渍采集进来形成噪声污点。为了除去指纹图像上的噪声点, 必须结合邻近的像素点的灰度值对它进行平滑处理。高斯滤波器是一种无论在空间域还是频率域都十分有效的低通滤波器, 能够有效地消除高频污染信号 (如噪声和细纹理等)^[28]。通常采用 3×3 和 5×5 大小的卷积模板, 效果如图 3-6, 为了避免过平滑造成的细节点丢失问题, 本文采用 3×3 高斯滤波模板进行卷积计算。

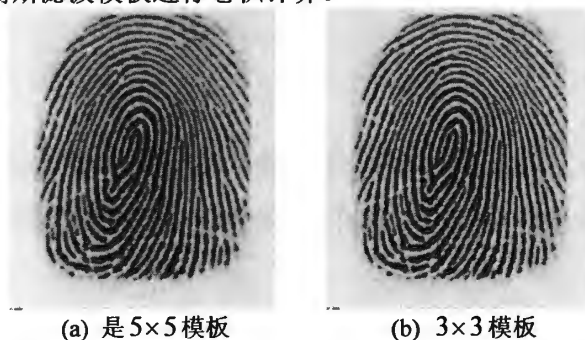


图3-6 高斯滤波效果图

3.5 指纹图像分割

指纹图像分为前景和背景两部分，前景就是有效的指纹对象，它包括脊线和谷线，是我们感兴趣的区域；背景是除有效的指纹对象之外的图像部分，包括不含指纹纹路的区域和由于噪声污染而无法恢复的低质量纹路区域。相对于前景区，背景区对于后续的分类、识别都没有太大意义，对其进行处理只会增加计算量。此外，对指纹图像进行分割可以为旋转校正提供全局轮廓，并通过剔除背景像素点提高指纹特征提取的正确率和计算速度，进而提升整个系统的性能，因此图像分割是指纹图像预处理过程中必不可少的一个步骤。

3.5.1 常见的图像分割方法

指纹分割常见的方法是基于灰度图像的方差法和基于方向图的方法^[19]。方差法是根据指纹图像背景区域的灰度变化少，因此灰度方差比较小，而前景区由黑白相间的纹理组成，因此方差比较大的特点进行分割。除了灰度分布信息外，方向信息也是指纹图像的重要特征。若某一区域是前景区，其脊线和谷线就具有相同的走向，经过统计后，在方向直方图中就会检测到峰值；而背景区域中各个方向出现的概率几乎相等，因此其方向直方图中无峰值出现。两种方法都能够近似完成分割，但因为都是以块为单位分割图像，导致前景区边界出现锯齿，不够平滑。若遇到受噪声污染的纹线区域，部分非背景区也会因为在噪声块中而被切割掉。

3.5.2 基于梯度场的图像分割方法

相对于前两种方法，以像素为单位的分割更为可靠，本文采用的是基于梯度场的分割方法。梯度场是一种反映指纹图像场的纹线隆起情况的物理场论。在指纹图像中，梯度是指指纹脊线隆起的程度，其方向是指纹纹线的法向方向，并从隆起指纹纹线横截面最高点指向最低点。对该点做偏微分，该偏微分的方向就是最陡方向，它是描述场强幅度变化的物理量^[27]。

综上，如果是指纹图像的前景区，因为指纹对象是黑白纹线交替出现，变化较多，所以其梯度场值较高；而背景部分并无明显的灰度变化，其梯度场值较低。如图 3-7，前景区域的梯度值明显不同于背景区。

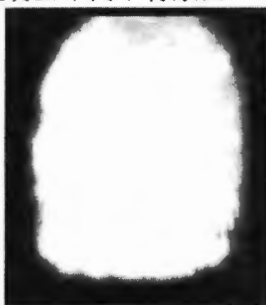


图3-7 指纹图像的梯度灰度化图

利用梯度场分割指纹图像的另一个优点就是：计算梯度场的中间结果可以直接用于后面方向场的计算，减小预处理时间复杂度。指纹图像的梯度计算过程如下：

首先，计算每个像素的水平梯度和垂直梯度， x 、 y 轴的偏导数使用梯度算子计算，常见的梯度算子有简单的 Sobel 算子、Prewitt 算子或是更加复杂的 Marr-Hildreth 算子。本文使用 Sobel 算子求偏导，算子如图 3-8 所示。

| | | |
|----|---|---|
| -1 | 0 | 1 |
| -2 | 0 | 2 |
| -1 | 0 | 1 |

x 轴方向

| | | |
|----|----|----|
| 1 | 2 | 1 |
| 0 | 0 | 0 |
| -1 | -2 | -1 |

y 轴方向

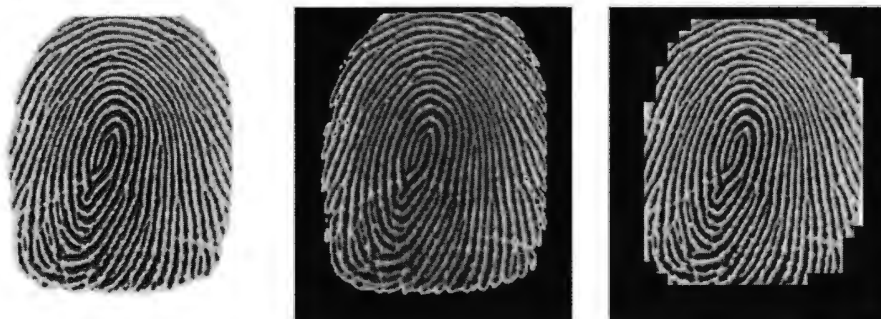
图3-8 Sobel 算子模板

用 G_x 、 G_y 表示每个像素点在 x 轴和 y 轴的偏微分量，其极坐标形式为： $G_x = \rho \cos\theta$ ， $G_y = \rho \sin\theta$ ，式中的 ρ 为梯度矢量的模， θ 就是以像素点 (i, j) 为中心的子块的局部脊线方向，准确地说，是与子块的 Fourier 频谱方向相垂直的方向。

由于平方和平方根需要大量的计算开销，经常使用绝对值近似表示梯度的幅值。

$$\rho = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y| \quad (3.8)$$

本文中的图像采集设备为电容式传感器，根据图像的平滑程度取分割阈值为 110.0。分割后指纹图像如图 3-9 的 (a)，相对于文献[19]中的灰度分割方法（效果如图 3-9 (b)），虽然此方法仍然需要人工确定分割阈值，但由于图像采集方式一致，设定的阈值具有较好的适用性，并且避免了图 (b) 中出现的误判情况；同时，相对于块分割方法（如图 3-9 (c)），前景区对象保持原有轮廓，边界线较为平滑。



(a) 本文改进算法的效果 (b) 常见的基于灰度分割的效果(c) 以块为单位分割的效果

图3-9 本文算法与文献中的算法对比示意图

3.6 指纹图像方向场的计算

Grasseli 在 1969 年首次提出了指纹方向场的概念，他将方向场定义为一个矩阵 D ，其元素为局部范围内脊线的切线方向^[15]。而指纹的方向图就是以指纹纹线的方向代替该位置灰度值的一种图像。

在后面的指纹图像识别过程中我们会采用方向场特征作为识别的一个重要要素，其优点有两个：首先，相比于细节特征点、脊线结构等指纹特征，指纹方向场特征一经获取，就会被多个处理步骤所使用，比如图像分割过程，图像增强

过程及特征提取过程等；其次，指纹图像方向场蕴含着丰富的特征信息，但其所占用的存储空间相对于原始的灰度图像却很小。例如，对于大小为 256×288 像素的指纹图像，按 15×15 大小进行分块，再将方向场角度量化为整数值，一幅指纹图像的块方向场仅需要约为 1.28K 的存储空间。

大多数的方向场计算方法都是基于像素之间的灰度关系^[29]。由于指纹图像大部分都是黑白间隔变化的纹线，图像梯度的方向就是它的灰度值变化方向，所以纹理的方向应该和图像梯度方向相互垂直^{[29][30]}。在实际应用中，还可以具体分为两种形式的方向场，一种是块方向场，主要是描述指纹图像中每个局部块中纹线的主导方向，以该方向表示该区域中所有像素点的方向；另一种是点方向场，主要是描述图像中每一个像素点所在的纹线的切线方向。两者比较起来，块方向场方法计算量较少，求解速度更快，并且考虑了整体方向的因素，但不利于后续求解奇异点；点方向场则具有更高的精度，更加准确地体现出奇异区方向变化快的特点，但容易受噪声影响，当图像质量不佳时，边缘像素的梯度方向并不垂直于纹路方向，会造成方向场计算错误。

3.6.1 基于梯度变化的块方向场计算

根据前文所述，我们可以直接通过梯度场的结果获得方向场。考虑到指纹方向取 θ 或 $\theta + \pi$ 时都是在一条直线上，对判断纹线走向没有影响，所以可以把角度加倍，将两者统一成一个方向，把方向场转换成一个连续的矢量场：

$$Re = G_x^2 - G_y^2 \quad Im = 2G_x G_y \quad (3.9)$$

1. 相应的极坐标公式如下：

$$Re = \rho^2 \cos 2\theta \quad Im = \rho^2 \sin 2\theta \quad (3.10)$$

2. 对方向图进行平滑，实部和虚部分别进行邻域平均：

$$V_{Re} = \frac{1}{N} \sum_w Re \quad V_{Im} = \frac{1}{N} \sum_w Im \quad (3.11)$$

式中 W 是 $w \times w$ ($w=15$) 的邻域模板，块大小的半径不能取得过大，一般为纹线的一个周期。关于如何选取窗口大小有一个经验方法：小半径能够准确描述图像的细节，而大半径则能够保留整体的轮廓，使纹线衔接的更为自然。但是，小窗口在增强图像细节的同时，也引进了大量的噪声，而大窗口则模糊了图像中的细节^[31]。

3. 最后，由于图像在空域的方向和在频域的方向是相互垂直的，所以获得了梯度场之后利用反三角函数就可以计算出方向场：

$$\theta = \frac{1}{2} \tan^{-1} \left(\frac{V_{Im}}{V_{Re}} \right) + \frac{\pi}{2} \quad (3.12)$$

$\frac{1}{2} \tan^{-1} \left(\frac{V_{Im}}{V_{Re}} \right)$ 的范围是 $[-\pi/2, \pi/2]$ ， θ 是最终的块方向，角度范围是 $[0, \pi]$ 。

实验证明，通过以上方法估计的脊线方向并不总是正确的。根据日常经验，在没有奇异点的邻域内，局部脊线方向是缓慢变化的，而在奇异区，脊线的方向变化是快速的。基于 Sobel 算子的块方向场方法因为要以主导方向代替块内方向，所以往往会模糊奇异点的方向细节，导致出现奇异点漂移的现象。奇异点漂移不仅会降低奇异点定位的准确性，还会影响到后续的基于方向场的指纹图像增强、二值化、细化等处理操作，最终造成纹线形态的改变^[32]。

综上, 本文在传统方法基础上对基于梯度的方向场计算方法进行了改进。

3.6.2 块方向场与点方向场相结合的方法

在求取块方向图的时候, 进行了多次分块加权平均处理, 实质上是利用数据相互扩散来抑制噪声, 由于扩散是双方面的, 这必然会带来另一个问题, 当错误方向的数据占到一定比例时, 方向场就会出错^[33]。为了让扩散变成单方向, 可以定量计算块方向一致性来帮助判断块方向场是否可靠以及是否能够正确描述奇异区。当块方向场不满足要求时, 用点方向来替换当前块内像素点的方向值。具体计算步骤如下:

1. 计算方向一致性

方向一致性反映的是指纹的局部方向变化程度, 因此, 本文针对块方向场, 通过将目标块与其 8 邻域块的方向角度做差, 计算目标块的方向一致性矩阵 $C(i, j)$, 方法如下:

$$C(i, j) = \frac{1}{N} \sqrt{\sum_{(i', j') \in D} |\theta(i', j') - \theta(i, j)|^2} \quad (i, j) \in \text{非背景块} \quad (3.13)$$

式中, D 表示块 (i, j) 的局部邻域 (3×3 大小), N 为 D 中非背景块的数目。 $\theta(i', j')$ 和 $\theta(i, j)$ 分别是块 (i', j') 和块 (i, j) 的局部方向场。在非奇异区, 如果在邻域 D 中所有的非背景块指向同一个方向, 则 $C(i, j)$ 趋近于 0, 即 $C(i, j)$ 越小表示方向越可靠。 $C(i, j)$ 的取值范围在 $[0, 1]$ 之间。

2. 块方向调整

块内的点方向调整需要满足两个条件:

A. 块内前景区像素点数大于块内像素点总数的一半时可以将其定义为前景块。为了防止非指纹区块数据对指纹区域数据扩散, 当进行方向一致性计算时只考虑前景块。

B. 当块 (i, j) 邻域内一致性系数 $C(i, j)$ 平均值大于阈值 T (本文 T 为 0.5), 说明此处方向变化较快, 则重新利用点方向方法计算这一块中每个像素点的方向, 以保留奇异区的图像细节。

计算指纹图像的方向场流程如图 3-10。

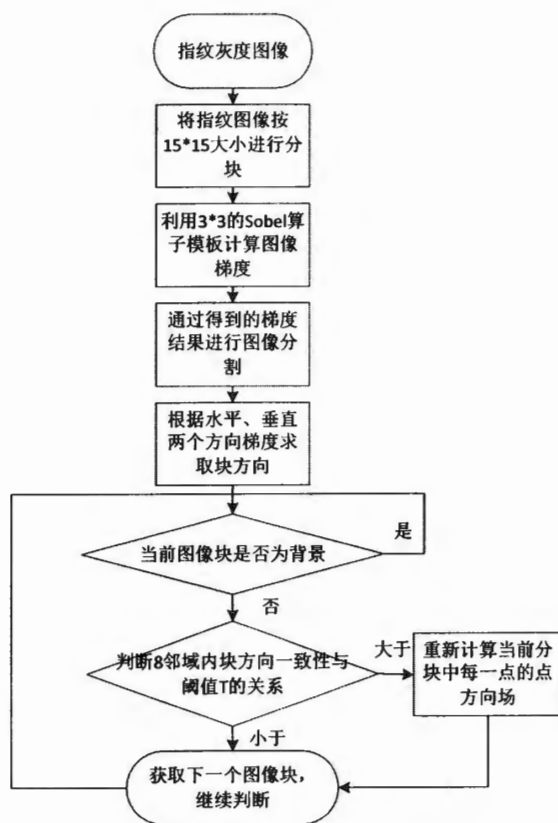


图3-10 求取方向场改进方法流程图

传统块方向场效果如图 3-11(a)，其计算简单，但精度差，易造成奇异点丢失；传统的点方向场如图 3-11(b)，在方向变化剧烈的奇异区和脊线谷线交替区域能够较好地保留方向信息，但计算速度慢，易被噪声干扰。

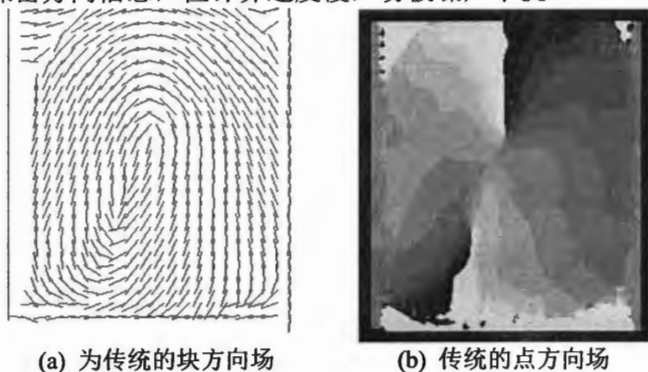


图3-11 块方向场和点方向场示意图

为了更清晰地展示改进算法的优势之处，本文将方向划分到 12 个方向和 4 个方向，图 3-12、图 3-13 为经过灰度化后的方向图，颜色越深角度越小，(b)图中可以看出奇异区附近已替换为点方向图，比(a)中的方向要清晰的多，既避免了噪声对指纹奇异点定位产生的不良影响，又保留了方向的真实变化。

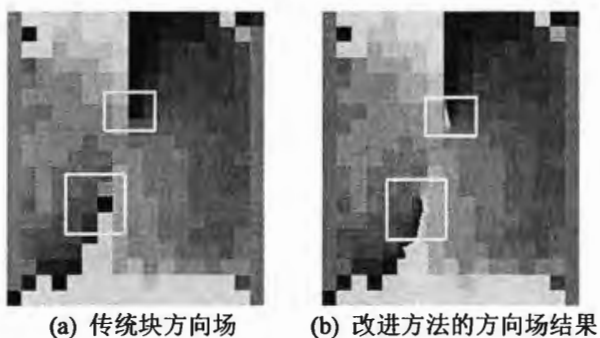


图3-12 12个方向的灰度化方向图

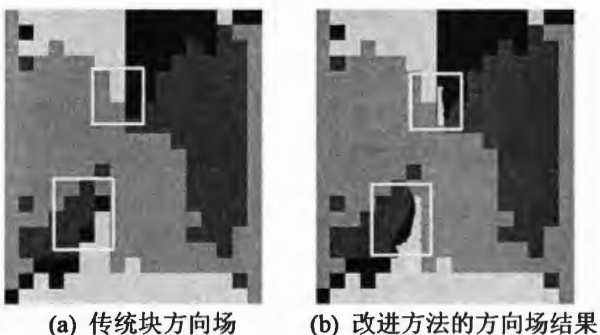


图3-13 4个方向的灰度化方向图

3.7 指纹图像增强

由于噪音压力等因素的影响,分割后的指纹图像往往存在纹线断裂或者粘连的现象,其中断裂会产生伪端点,粘连会产生伪分叉点。为了避免这两种噪音带来的虚假特征点,必须在特征提取之前采取措施,避免影响指纹的识别工作。因此,指纹图像增强是预处理算法中不可或缺的一步。

指纹图像的增强主要是根据指纹纹线的方向进行滤波,参照不同的角度设计相对应的滤波器,以达到对指纹纹线的断裂进行连接和纹线粘连进行断开的效果。常见的增强算法主要分为 Gabor 滤波器算法和方向模板滤波方法。

3.7.1 Gabor 小波滤波器

Gabor 变换是由英国物理学家 Gabor 提出,其函数由高斯函数和三角函数构成,形成周期振荡的函数^[34]。Gabor 滤波器由独立的偶对称滤波器和奇对称滤波器组成,前者用于目标检测,后者用于边缘检测。另外, Gabor 滤波器结果的实部和虚部分别对应于偶对称滤波器和奇对称滤波器的滤波结果^[30]。考虑到本文是为了增强图像而不是分割图像,所以只需要采用偶对称即可,其数学表达式为:

一维表达式为:

$$H(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{x^2}{2\sigma^2}\right\} \cos(2\pi fx) \quad (3.14)$$

二维表达式为:

$$H(x) = G(x', y') \cos(2\pi fx') \quad (3.15)$$

公式的具体说明如下:

高斯分量:

$$G(x, y) = \frac{1}{2\pi\lambda\sigma^2} \exp\left\{-\frac{(x/\lambda)^2 + y^2}{2\sigma^2}\right\} \quad (3.16)$$

坐标轴旋转公式:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \sin\theta & \cos\theta \\ \cos\theta & -\sin\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (3.17)$$

式中 f 是指纹图像的频率场, 在实际的增强过程中, 为了减少计算量, 通常采用整幅指纹图像的一个主要频率来代替每一块指纹的具体频率, σ 是高斯滤波器沿 x 轴和 y 轴的空间常量, λ 是坐标轴比例因子, 默认为 1, θ 是相应像素点的方向。

在实验中, 根据经验和图像的实际统计值, f 近似取值为 7, σ_x 取 4.0, σ_y 取 3.0, 效果较好。同时, 为了提高 Gabor 模板的生成效率, 可以根据与 x 轴的夹角 θ 产生十二个方向的滤波器模板, θ 取值为 $0^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ, 75^\circ, 90^\circ, 105^\circ, 120^\circ, 135^\circ, 150^\circ, 165^\circ$ 。将所有参数带入一般的 Gabor 滤波器公式, 以 $\theta = 0^\circ$ 为例, 表 3-1 就是求得的相应滤波器模板:

表3-1 Gabor 滤波器的水平方向模板系数

| | | | | | | |
|---------|---------|---------|---------|---------|---------|---------|
| -0.3314 | -0.4376 | -0.5169 | -0.5465 | -0.5169 | -0.4376 | -0.3314 |
| -0.1081 | -0.1427 | -0.1686 | -0.1782 | -0.1686 | -0.1427 | -0.1081 |
| 0.3577 | 0.4722 | 0.5579 | 0.5898 | 0.5579 | 0.4723 | 0.3577 |
| 0.6065 | 0.8007 | 0.9460 | 1.0000 | 0.9460 | 0.8007 | 0.6065 |
| 0.3577 | 0.4722 | 0.5579 | 0.5898 | 0.5579 | 0.4723 | 0.3577 |
| -0.1081 | -0.1427 | -0.1686 | -0.1782 | -0.1686 | -0.1427 | -0.1081 |
| -0.3314 | -0.4376 | -0.5169 | -0.5465 | -0.5169 | -0.4376 | -0.3314 |

如果使用 Gabor 滤波器, 每一个像素点都要通过以上表中的模板 (或者旋转后的模板) 进行滤波计算, 其时间消耗是非常大的, 所以需要使用更为简单的方向滤波器。

3.7.2 结合 Gabor 滤波器特性的十字方向滤波方法

受到 Gabor 滤波器的启发, 针对指纹图像脊线被破坏的两种情况, 在得到指纹图像的方向图后, 可以直接将每个像素点的灰度值卷积相对应的方向滤波模板, 从而连接断裂纹线, 去除粘连现象。方向滤波器的方法如下^[35]:

1. 为了令 x 轴和 y 轴相对于中心点对称, 模板的尺寸应为奇数, 并且大小要合适, 一般是半个纹线周期 (本文取为 7)。

2. 根据 Gabor 函数计算的模板规律, 为了提高脊线和谷线的灰度差异, 纹线的切线方向上应该都为正, 并且系数大小相同, 而纹线法线方向上的边缘系数应该为负。

3. 根据滤波器本身设计的特性, 滤波结果与原始图像的平均灰度无关, 因此, 所有系数的代数和应该为 0。

| | | | | | | |
|---|---|---|---|---|---|---|
| W | W | W | W | W | W | W |
| Z | Z | Z | Z | Z | Z | Z |
| Y | Y | Y | Y | Y | Y | Y |
| X | X | X | X | X | X | X |
| Y | Y | Y | Y | Y | Y | Y |
| Z | Z | Z | Z | Z | Z | Z |
| W | W | W | W | W | W | W |

图3-14 7×7方向滤波器

综上，以图3-14为例，这是一个7×7的方向滤波器模板。竖直方向上是分离滤波器，其系数满足： $X \geq Y \geq 0 \geq Z \geq W$ 。如果图像中出现粘连，则经过分离滤波器处理后，粘连点的灰度值增加，减弱了与上下两行像素点的关联。而方向滤波器的水平方向是一个均值滤波器，其作用就是令当前像素点的灰度值调整为相邻点的均值，连接断裂的纹线。

通常，滤波器是 $N \times N$ 大小的矩形模板，虽然计算量较 Gabor 函数有了一定的缩减，但仍然无法满足实际系统的要求。本文参考文献[36]中的十字滑动窗口的思想，将设计好的矩形模板改进成两条交叉的十字模板。如此一来，计算量较原来减少了 73%。具体的计算方法如下：

首先对前景区的每个像素点进行纹线切线方向的均值滤波，然后再进行纹线法线方向的高斯滤波。以 $\theta = 0^\circ$ 为例，如图 3-15：

| | | | | | | |
|---|---|---|---|---|---|---|
| X | X | X | X | X | X | X |
| K | | | | | | |
| Z | | | | | | |
| Y | | | | | | |
| X | | | | | | |
| Y | | | | | | |
| Z | | | | | | |
| K | | | | | | |

(a) 水平方向模板 (b) 垂直方向模板

图3-15 十字交叉滤波模板

滤波时，每一点的灰度值是由自身和邻域共 12 个点的灰度值共同决定。模板系数可以通过计算 Gabor 滤波器的模板简化获得。经过实验验证， $X=9$ ， $Y=3$ ， $Z=-1$ ， $K=-3$ 。最终的模板如图 3-16。

| | | | | | | |
|----|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| -3 | | | | | | |
| -1 | | | | | | |
| 3 | | | | | | |
| 9 | | | | | | |
| 3 | | | | | | |
| -1 | | | | | | |
| -3 | | | | | | |

(a) 水平方向模板 (b) 垂直方向模板

图3-16 十字交叉滤波模板系数

本文设计了 8 方向和 12 方向的十字滤波器，在得到 0° 方向的滤波模板后，其他方向的滤波器均可以由其按公式 3.18 的方法旋转得到，旋转后，滤波器上坐标 (i', j') 与 0° 方向滤波器上坐标 (i, j) 的关系见公式 3.18：

$$\begin{bmatrix} i' \\ j' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} \quad (3.18)$$

其中, $\theta = l\pi / N$ (N 为 8 或者 12, $l = 0, 1 \dots N$) 为滤波器旋转的角度。

设计好滤波器的模板后, 在指纹图像上, 按照顺序依次根据每一个像素点的方向值选取相应方向的滤波器掩码模板, 再利用公式 3.19 和公式 3.20 进行卷积滤波运算:

$$f'(i, j) = \sum_{u=-3}^3 f(i+u, j) g_{\theta}(u) \quad (3.19)$$

$$f''(i, j) = \sum_{v=-3}^3 f'(i, j+v) g_{\theta}(v) \quad (3.20)$$

其中, $g_{\theta}(u)$ 和 $g_{\theta}(v)$ 分别是相应方向滤波器水平和垂直掩码模板的系数。实验结果对比如图 3-17。



图3-17 增强效果对比示意图

通过图 3-17 可以看出, Gabor 滤波具有很好的自适应性, 经过处理后, 指纹图像中的一些噪声被滤除了, 指纹中存在的断纹也被连接上了, 但同时也误连接了一些本应断裂的部分, 这样反而会影响特征提取的判断。为了更加清晰地对比结果, 将增强后的图像 (c) 进行二值化获得图 (d), 与图 (b) 比较可以看出其指纹脊线相对更加的平滑, 为下一步操作中的细化提供了清晰的参考图像。

总体来说, Gabor 滤波器从本质上来说仍然是一种方向滤波器, 但是其考虑的因素较多, 卷积的模板范围较大, 纹线频率的计算又相对比较复杂和耗时, 很容易因为外界噪声而产生很大的偏差, 即使采取固定脊线频率的方法, 7×7 的模板滤波计算仍然无法接受。相反, 十字交叉滤波器不但计算量大大减少, 而且还能获得较好的增强效果, 因此, 本文采用此方法进行指纹图像增强处理。

3.8 指纹图像二值化

3.8.1 二值化

图像二值化就是将一幅 256 级灰度图像转换成只有黑白两级的图像。其中，0 表示相对色调较深的脊线，255 表示较浅的脊线，处理后，图像灰度层次也从原来的 256 级减少到 2 级，相应地，图像的像素深度也从 8bit 减少到 1bit。二值化处理的目的是在保留指纹原图的细节特征的基础上突出原始图像的纹路，并且减小存储空间的使用。常用的二值化方法有全局阈值法，局部阈值自适应法和方向二值化法等^[37]。

本文采用文献[29]中基于指纹纹线方向的二值化方法：根据指纹纹线的分布特点，对于脊线上的深色点来说，沿脊线法线方向上的像素点应该都是浅色点，其灰度值之和较大；而切线方向的像素点应该都是深色点，其灰度值的和较小。对于谷线，其规律正好相反。由此分别对纹线方向和垂直方向的灰度值进行加权求和，根据大小关系确定当前像素点在脊线上还是谷线上。

图 3-18 是 0° 方向时的加权模板，其他 11 个模板可以通过公式 3.18 旋转获得，图 3-19 是基于灰度和基于方向信息两种方法的结果对比图。

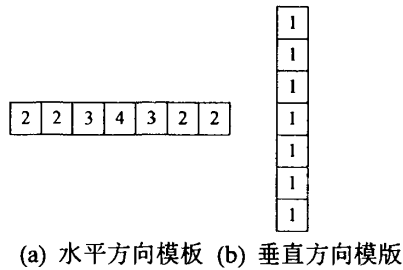


图3-18 二值化灰度加权求和模板



(a) Otsu 法^[34]的二值化结果 (b) 基于方向信息的二值化结果

图3-19 本文使用的二值化方法与文献中方法的对比示意图

实验结果表明，基于指纹方向信息的二值化方法基本上可以保证重要细节信息不被破坏，纹线也相对更加平滑自然。

3.8.2 二值化去噪

二值化后的脊线中间有时会出现小孔，如果不及时去除这些噪声而直接进行细化的话，这些小孔就会变成错误的分叉，或形成岛形和桥形结构，为特征提取

带来了挑战。通常，这些伪特征很难在特征提取阶段去除，这又给匹配识别造成了困难^[38]。一种常用的二值化去噪方法可以较好地消除二值化后的小孔：

- 1. 把满足以下两个条件的像素点灰度值取 0：
 - P 灰度值为255。
 - P 的四邻域中有3个以上的邻点灰度值为0。
 - 2. 把满足以下两个条件的像素点灰度值取 255：
 - P 灰度值为0。
 - $((P_1 + P_2 + P_3) \parallel (P_5 + P_6 + P_7)) + ((P_3 + P_4 + P_5) \parallel (P_7 + P_8 + P_1)) = 255$ 。
- 其中 $P_1 \sim P_8$ 位置的定义如图 3-20。

| | | |
|----|----|----|
| P4 | P3 | P2 |
| P5 | P | P1 |
| P6 | P7 | P8 |

图3-20 二值化去噪模板

最终的二值化结果如图 3-21：



图3-21 二值化去噪效果图

3.9 指纹图像骨架的细化提取

3.9.1 细化

指纹图像二值化后，其纹线粗细是不均匀的，这不利于提取细节特征。细化就是在不影响纹线连通性的情况下，将纹线宽度缩窄到一个像素^[23]。

指纹图像细化方法有很多，本文主要考虑到图像处理的实时性要求，采用速度较快的查表法^[39]。

根据文献[39]里的总结：(1)内部点不可以删除；(2)孤立点不可以删除；(3)直线端点不可以删除；(4)如果当前点在边界，去掉后若连通分量并不增加，则可以删除。根据以上这四个判断条件，事先准备一张表，从 0 到 255 共有 256 个元素，每个元素要么是 0，要么是 1。根据当前点的八邻域点计算查表索引，若表中的元素是 0，则保留该点，否则删除。

3.9.2 细化去噪

细化去噪处理是为了减少细化图中的伪特征而做的一些处理工作。虽然在二值化后对噪声进行了一些去除，但细化算法仍会带来噪声，导致特征提取时得到错误的特征点。因此，对指纹图像细化后，需要尽可能地消除潜在的噪声，保留真正的指纹细节。

本文采用文献[39]中的去噪方法，图 3-22 中的 (a) 图是去噪之前的细化图像，(b) 是经过去噪操作后的结果图。

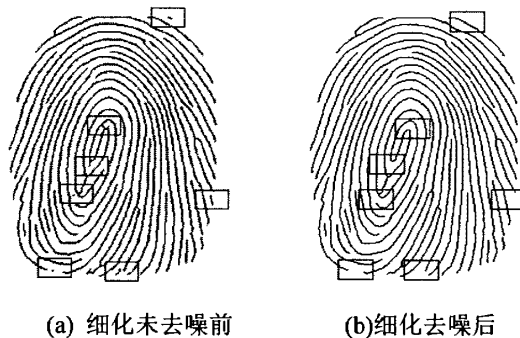


图3-22 细化未去噪与去噪后结果的对比图

上图显示，细化后的图像存在一些毛刺和短线等噪声，采用去噪处理后伪特征都已经被剔除。

3.10 本章小结

本章深入讨论了自动指纹识别系统的预处理过程，完成了预处理流程中的图像直方图均衡化、归一化、图像平滑、图像分割、方向图计算、图像增强、图像二值化及去噪、图像细化及去噪等工作，并且根据电容式传感器采集到的指纹图像的特点，主要对直方图均衡化、方向场计算以及图像增强进行了改进。包括在传统直方图均衡化基础上利用分段函数确定概率统计范围，结合点方向场能够保留奇异区特征和块方向场运算速度快的优点计算方向场，利用十字方向滤波模板对指纹图像进行增强，快速消除粘连和断裂噪声三个方面。实验结果表明此套算法具有可行性和一定的自适应性，能够准确处理大部分被噪声污染的指纹图像，为后续工作提供细化彻底的二值化图像。

第四章 指纹图像多级分类

指纹分类是指在指纹图像经过预处理后,通过一定的分类特征将其划分为某一类别中的过程。经过分类后,就可以在缩小的类别集合中完成指纹的匹配。对于一般小规模指纹识别系统,由于用于指纹分类的训练样本较小,分类准确率很难达到 100%,预先所做的分类工作很可能没有节省时间,反而因为提取分类特征和决策类别减慢了指纹匹配的速度。但对于一个大规模的应用来说,分类技术可以加快检索的速度,大幅度减少特征比对的时间,提高整个系统的性能。综上所述,指纹分类技术能够为大规模指纹数据库提供重要的检索机制,以此加快识别的执行速度,是自动指纹识别系统的重要环节之一,具有一定的研究价值。

4.1 指纹分类基础

指纹本身具有可分类性,这为指纹分类提供了理论基础。指纹分类的研究主要关注如下几个方面^[19]:

1. 指纹的类别数目。通常,类别数目越多,子样本集就越小,匹配的 efficiency 就越高。但在实际的指纹分类系统中,类别分得过细反而降低了分类的准确率,增加了分类的拒识率和误识率。另外,指纹各个类别类间相似性大,类内相似性小,很多指纹图像往往可以同时属于几个类别,连人工都很难准确判断。

2. 指纹分类尽量均匀。指纹类别分布均匀有助于提高平均查找速率,降低穿透率。事实上,在自然条件下,指纹各个类型的分布并不是均匀的。据统计,90%以上的指纹是属于斗型、左箕型和右箕型三种类型之一。所以目前还无法做到真正的均匀分类。

3. 分类准确率高、拒绝率低。分类的准确率必须达到较高的水平,否则匹配时就无法比对正确的样本子集,分类将没有任何意义。同时,要正确选择用于分类的特征,防止因为缺少某个分类特征而被拒绝分类。

4. 分类算法的时间复杂度。指纹分类模块加入的目的就是为了提高指纹识别系统的性能,如果加入这一机制后的搜索速度反而比正常的线性搜索还慢,那么研究指纹分类的意义就不大了。

4.2 指纹分类方法性能评估

当指纹类别数目固定的情况下,常是以分类准确度和时间复杂度作为衡量指纹分类方法性能的核心指标^[40]。而分类的准确度主要是利用系统的穿透率 P (Penetration Rate) 进行评估,其计算公式如 4.1:

$$P = \frac{\text{子样本中匹配的指纹数目}}{\text{指纹库中的指纹总数}} \times 100\% \quad (4.1)$$

P 的范围是 $\left[\frac{1}{N}, 1\right]$, 穿透率越小表示分类后需要匹配的样本范围越小。如果不分类直接进行线性比对搜索,那么系统穿透率是 1。所以,指纹分类就是为了缩小匹配范围,降低穿透率 P 。

更为直观和具体的评估方法是利用混合矩阵,其每一列描述的是正确的指纹

类型，每一行是描述的是通过实验所获得的分类结果，第 i 行 j 列的值表示 j 类型的指纹被分到 i 类型的个数^[15]。本文实际采用的是混合矩阵来评估分类的准确率。

4.3 常见的指纹分类方法

大量的参考文献表明，使用最多的指纹分类方法是 Henry 分类方法^[40]，主要将指纹分为斗(Whorl)、左簸箕(Left Loop)、右簸箕(Right Loop)、双旋(Twin Loop)、尖拱(Tented Arch)五类。由于这种分类方法存在一个缺点，即五种类型的指纹分布概率十分不均匀，所以在此基础上出现了很多变种方法，进一步提高分类结果准确性。

目前，针对分类的方法有很多，根据分类思想不同，可以分为基于知识的方法、基于结构的方法、基于句法分析的方法、基于滤波器的方法等。根据分类特征的不同，又分为基于指纹结构信息的方法、基于点方向信息和块方向信息的方法以及基于指纹纹线编码信息的方法^[15]。其中，基于指纹结构信息的方法属于模型方式，主要利用指纹全局特征点的数目和相对位置进行分类，这类方法更加接近人类的思维习惯，是一种“智能化”的方法；后三种则属于全局方法，是根据整个图像方向场或脊线结构特征进行分类，常采用统计学原理方法^[41]。

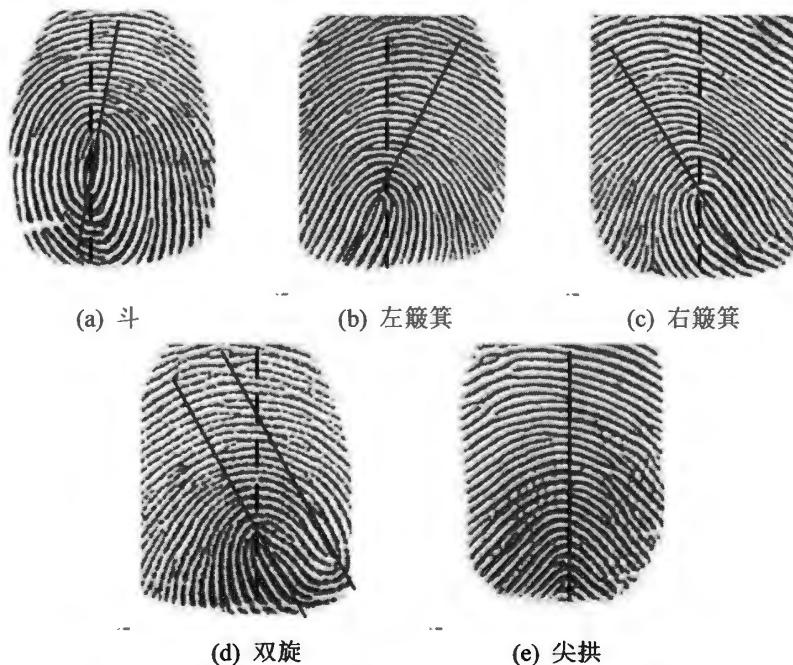


图4-1 在五种类别的指纹图像上标记对称轴的示意图

考虑到本文的研究对象是民用指纹识别系统，采集到的指纹图像相对质量较好，所以可以采用基于知识的分类方法。在对指纹图像的分类处理操作中，我们只对具有奇异点的中心区域感兴趣，以 Henry 分类法的五类为例，观察图 4-1 可以看出，方向摆正的指纹图像存在非常明显的对称性，其对称轴如图中红色实线标注，竖直轴如黑色虚线标注，通过比较对称轴方向与竖直方向的夹角就能帮助实现指纹分类。

由此说明，在分类之前需要对指纹图像进行一些分类预处理。然后，在常规预处理的图像分割步骤后，需要对旋转的图像进行校正，再根据指纹图像的方向场，准确提取用于分类的中心点特征等，最后设计具体的分类算法。指纹分类算法工作过程如图 4-2 所示。

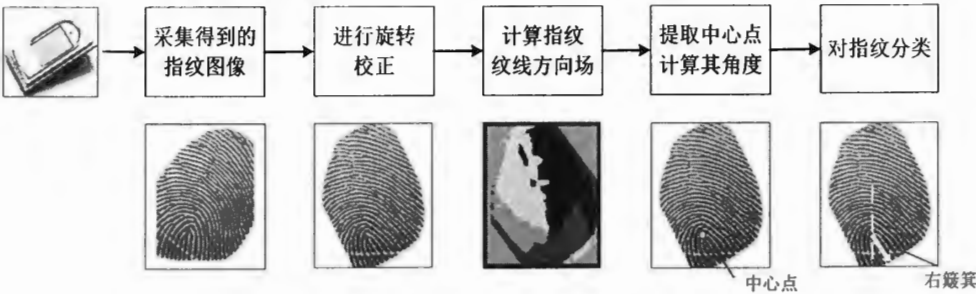


图4-2 指纹分类算法过程

4.4 指纹旋转校正

4.4.1 校正指纹图像的必要性

在采集指纹图像时，不同的人放置手指会存在不同程度的差异，即使是同一枚手指在不同时刻的采集过程中也会发生旋转、拉伸等情况。这其中，由于拉伸造成的变形只是会轻微影响到局部特征，一般忽略不计，而旋转则分为俯仰（pitch）、航向（yaw）和横滚（roll）^[42]三种。



图4-3 三种旋转方式的定义

俯仰是将实物绕 X 轴旋转（Local Rotation X），航向是将实物绕 Y 轴旋转（Local Rotation Y），横滚是将实物绕 Z 轴旋转（Local Rotation Z），三种旋转方

式的定义方式如图 4-3 (a)。俯仰旋转和横滚旋转会造成采集面积减小, 分别只采集到了指纹的上(或下)部分和指纹的左(或右)部分, 如图 4-3 (b)、(c) 所示, 而航向旋转只是改变指纹图像整体轮廓的全局方向和位置, 图像的纵向对称轴偏离竖直方向, 指纹的有效面积是与无旋转时近乎无差别的, 如图 4-3 (d)。在实际应用中, 当指纹的有效面积无法达到要求时, 需要提示用户重新采集, 所以本文重点处理航向旋转。

虽然为了实时性应当尽量精简对指纹图像的处理步骤, 但是考虑以下三方面, 本文有必要进行旋转校正:

第一, 为了指纹分类和指纹细配准, 通常我们需要定位奇异点, 而从采集到的奇异点集中剔除伪点的规则是建立在指纹图像是摆正的基础上的。

第二, 如果指纹图像的中心对称轴是处于竖直方向, 则可以直接根据对称轴方向和 y 轴方向的夹角对指纹进行粗分类, 再以螺径、箕型拱形出口方向等继续细分类, 能够显著提高指纹匹配时的搜索速度。以图 4-4 (a) 为例, 其对称轴虽然是近似竖直方向, 但实际上指纹是右簸箕, 而图 4-4 (b) 则正好相反, 其实际上是左簸箕, 但因为整体图像的旋转, 对称轴偏向右侧, 容易被误分类为右簸箕。



图4-4 (a)中指纹为右簸箕图像, (b)为左簸箕图像, 因发生了旋转无法准确得到种类

第三, 在指纹样本库中保存的指纹图像都是清晰完整、无旋转的, 旋正的指纹在匹配过程中就无须大范围地旋转配准, 可以方便地对两幅图像进行平移对齐。

4.4.2 文献中的旋转角度确定方法

在文献[15]中, 作者首先分析了奇异点附近的方向分布规律具有对称性, 然后分割出指纹的模式区, 统计其方向场直方图, 通过平移的距离值确定图像需要旋转的角度。但文中也同时指出, 这种方法具有很大的误差, 需要不同的策略来进行抵消, 计算量较大。

Cao 和 Yang 等^[43]根据指纹图像通常呈现细长矩形这一特点, 首先提取指纹图像前景区域的整体轮廓, 然后计算包含该轮廓的最小外接凸多边形, 最后再根据凸多边形确定多个外接矩形, 选择包含该凸多边形的所有矩形中面积最小者。鉴于指纹放置一般向上放置, 所以长边的上行方向被视为手指放置的倾斜方向。整个过程如图 4-5, 其中指纹图像来自于 FVC2004 DB1 指纹库。

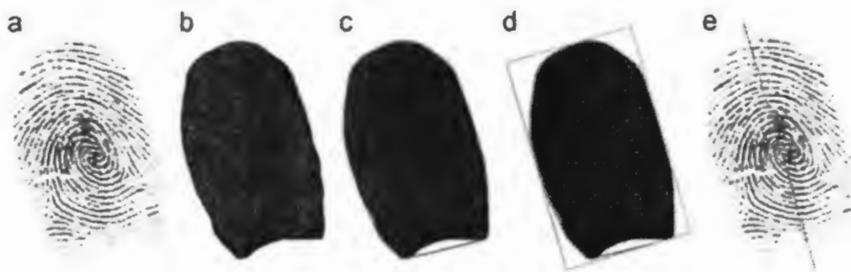


图4-5 文献[43]中手指放置方向计算过程图示

此方法虽然符合指纹规律，且算法实现简单，但是需要指纹采集仪的传感器面积足够大，如此才能够保证采集到的指纹轮廓是呈现细长形状的。本文所使用的采集仪采集面积仅有 256×288 个像素，所得到的指纹图像的长宽差异不大，以图 4-6 中的指纹为例，该算法很难确定外接矩形的长边是竖直方向还是水平方向。



图4-6 被误判的旋转指纹

通常，指纹轮廓是近似轴对称的图像，如果图像发生旋转，竖直方向对称轴偏转的角度就是整个指纹图像旋转的角度，文献[44]根据这一规律，描述了一种拟合前景区中位点，通过计算所得直线的斜率获得倾斜角度的方法，但应用于实际情况时，会出现对称轴拟合存在误差导致角度不正确的情况。图 4-7 中指纹图像的左右两侧因为采集头的限制而被切割，并未完全地展现指纹的全局轮廓，造成定位的中位点并非是正确的点。其中红色直线是正确的对称轴，黑色直线是拟合所得的对称轴。该方法的实验结果与 Cao 和 Yang 的方法相反，出现欠校正的情况，即校正度数不够或无法判断是否发生偏转。



图4-7 文献[44]确定旋转角度的方法图示

4.4.3 本文改进的方法

在对比了相关文献的方法后，本文将吸取多种方法的优点，综合确定最终的旋转角度。首先，为了去除俯仰旋转和横滚旋转带来的影响，根据之前获得的前

景图像，计算指纹有效区域占整幅图像的比重，如果小于一定的比例，则可以认为有效面积太小，无需进行后续的工作。

鉴于本文所研究的指纹图像采集工作是可重复的，并且没有人为不配合的情况，所以指纹不会偏转较大，可以将指纹图像较竖直方向的旋转范围设定在 $[-90^\circ, 90^\circ]$ ，如图 4-8 所示，当旋转角度为正时本文定义为右旋，反之为左旋。如果有更大的旋转，则可以认为用户没有按照要求放置手指，需要提示重新采集。



图4-8 指纹旋转角度的范围定义示意图

经过 3.5 节对指纹图像进行分割后，我们得到了如图 4-9 (b) 的前景图像，并且定义前景区的正外接矩形的高为 Height 和宽为 Width。

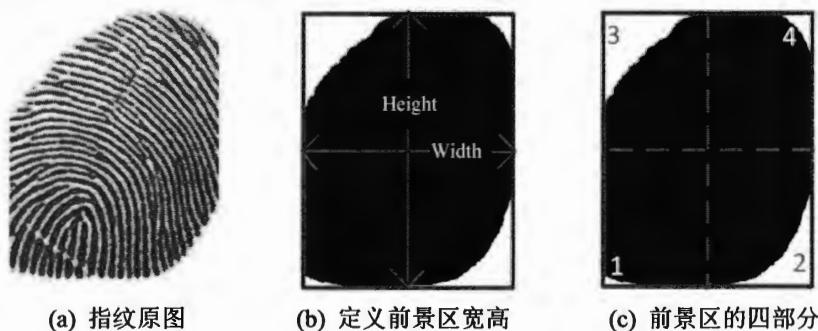


图4-9 对前景区分区的示意图

通过观察大量的旋转指纹图像可以发现，旋转的指纹有一个重要的特征：如图 4-9 (c) 中标识的方式将指纹图像的最大外接矩形划分成 1、2、3、4 标注的四个区域，以 $pixels_1$ 、 $pixels_2$ 、 $pixels_3$ 、 $pixels_4$ 表示对应的背景点数，若指纹发生了偏转，图 4-9 (c) 中获得的前景区外接矩形必然会存在某两个对角线对称区域有相对较多的背景点，而另外两个区域存在较少背景点的情况。例如：(2, 3) 区域中灰度值为 255 的像素点明显高于 (1, 4)，则可以粗略地认为是一幅左旋图像。所以，计算旋转角度的第一步就是统计四个部分的像素点，对旋转方向进行预判。

然后，采用文献[44]中拟合中位线的方法，先是计算指纹前景图每一行的中心点，接着，使用 OpenCV^[45]提供的 `cvFitLine()` 函数将这些离散点拟合成直线。

```

void cvFitLine(
    const CvArr* points, //points是2D待拟合的点集
    int dist_type, //dist_type取CV_DIST_L2 , 代表最小二乘法
    double param,
    double reps, //半径精度
    double aeps, //角度精度
    float* line //拟合结果
);

```

结合函数所返回的 line, 可以根据直线斜率的反正切结果获得旋转角度。如图 4-10, 中间点集 (Middle points group) 基本分布于对称轴的两侧, 在建立了旋转对称轴后, θ 就是所求角度。

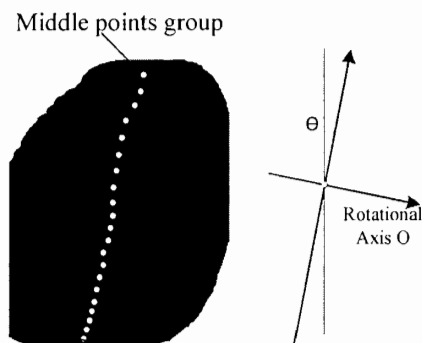


图4-10 拟合中位线方法示意图

接着, 对文献[43]中的方法进行改进, 无需计算外接凸多边形, 直接对轮廓进行检测。针对包含边界曲线的最小矩形并不总是正确表示指纹倾斜角度的矩形这一问题, 本文使用 OpenCV 自带函数 `cvFitEllipse2()` 和 `cvMinAreaRect2()` 找到最小外接面积的椭圆和矩形, `cvFitEllipse2()` 的功能是对给定的一维点集作椭圆的最佳拟合 (基于最小二乘法)。返回的数据结构中包含椭圆轴的整个长度, 椭圆旋转的角度等参数。

```

CvBox2D cvFitEllipse2(
    const CvArr* points, //points是点集的序列或数组
);

```

`cvMinAreaRect2()` 的功能是通过建立凸外形并且旋转外形以寻找给定 2D 点集的最小面积的包围矩形。返回的数据结构中同样包含矩形的旋转角度。

```

CvBox2D cvMinAreaRect2(
    const CvArr* points, //points是点序列或点集数组
    CvMemStorage* storage = NULL //storage是可选的临时存储仓
);

```

同样如图4-11, 建立了旋转坐标系, 根据长轴倾斜度获得角度 θ 。

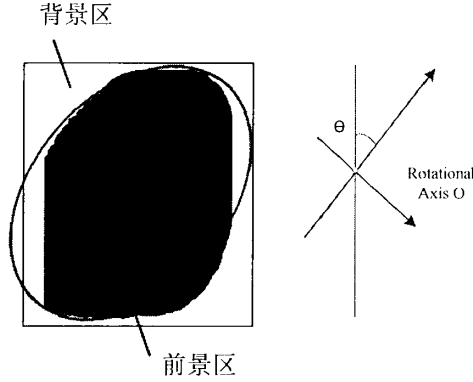


图4-11 外接椭圆法示意图

最后，我们定义椭圆的旋转角度为 $\theta_{ellipse}$ ，长轴为 b ，短轴为 a ，矩形的旋转角度为 θ_{rect} ，中位线的倾斜角度为 θ_{line} ，需要校正的角度为 θ ，综合两种方法的结果进行判断：

A. 根据经验，矩形若存在倾斜角，则原始指纹图像一定发生了旋转，所以当 $|\theta_{rect}| > 25^\circ$ 时，直接跳至判断条件 D，否则继续判断条件 B；

B. 多次试验结果表明，外接椭圆为了包含整个指纹对象，旋转的角度往往大于实际指纹的角度，所以会出现过校正的情况，所以当 $|\theta_{ellipse}| < 7^\circ$ ，可以直接断定指纹不需要校正，即 $\theta = 0^\circ$ ，否则继续判断条件 C；

C. 若满足 $|\theta_{line}| < 6^\circ$ ，角度虽然很小，但不足以说明图像有没有偏转，为了进一步验证，还需要利用外接椭圆的相关参数：如果满足 $|\theta_{ellipse}| < 15^\circ$ 或 $|b/a| < 0.9$ 两个条件，则可以认为 $\theta = 0^\circ$ ，直接输出图像，否则继续判断条件 D；

D. 有两种情况拟合直线法无法捕捉到正确的旋转方向，可以直接令 $\theta = \theta_{ellipse}$ ：一是当指纹旋转角度绝对值超过 45° 时，二是两种方法计算的结果正负相反并且绝对值相差太大时，其他情况可以通过公式 4.2 计算 θ 。为了弥补两种方法的欠校正和过校正的情况，本文按照 $\theta_{ellipse} : \theta_{line} = 6:4$ 的比例计算最终的旋转角度：

$$\theta = \theta_{ellipse} * 0.6 + \theta_{line} * 0.4 \quad (4.2)$$

旋转角度计算的流程图如图 4-12。

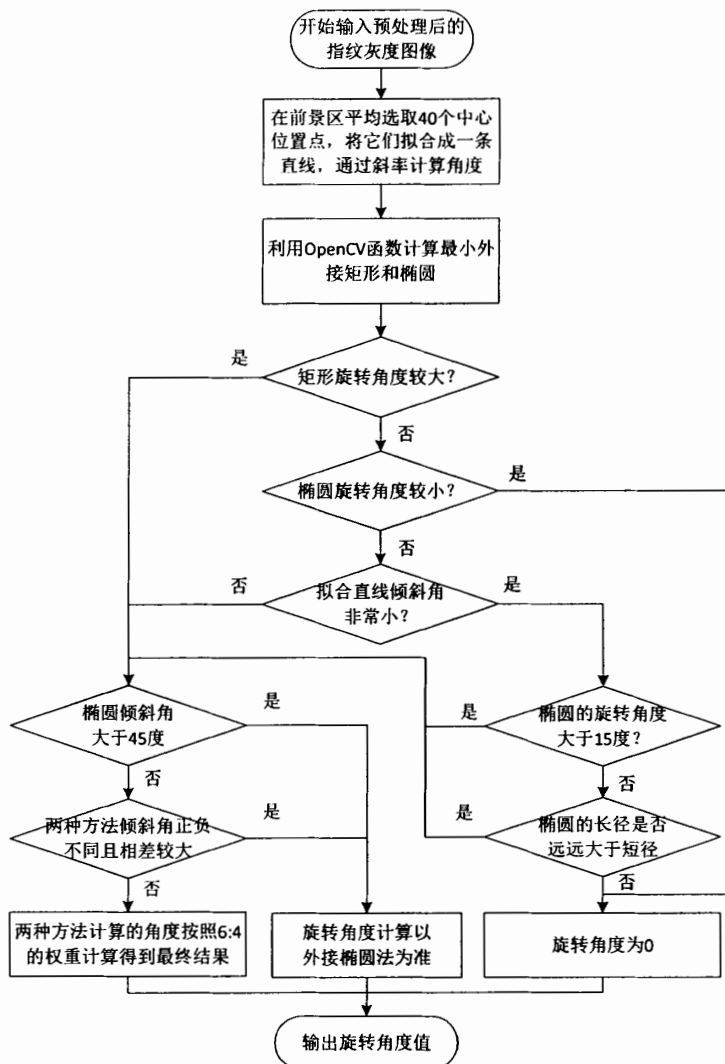


图4-12 旋转角度计算流程图

4.4.4 依据旋转角度旋转图像

在已知指纹图像是否旋转以及旋转的角度后，接着就是要将指纹图像进行几何变换使其对称轴处于竖直方向。文献[46]中为了避免灰度图像的插值运算而仅仅旋转方向图。依据旋转不变性，指纹局部特征点的相对位置、相对的旋转角度虽然是不变的，但提取奇异点和指纹的匹配操作中仍然需要指纹图像的绝对特征。而且仅仅将旋转角度限定在 22.5° 和 45° ，不能获得相对准确的校正图像，所以本文采用数字图像处理中常用的仿射变换算法对灰度图像校正。

对一幅图像做转换，就意味着要进行一些插值运算以使输出的图像平滑并且看起来自然一些。OpenCV中为稠密变换提供的转换函数是 `cvWarpAffine()`：

```

void cvWarpAffine(
    const CvArr* src, //原始图像或者数组
    CvArr* dst, //目的图像或者数组
    const CvMat* map_matrix, //对所需要的变换进行量化矩阵
    int flag = CV_INTER_LINEAR |
    CV_WARP_FILL_OUTLIERS,
    cvScalarAll fillval = cvScalarAll(0) //默认
);

```

参数 `map_matrix` 由OpenCV中提供的 `cv2DRotationMatrix()` 求得, 它用来计算围绕任意点的旋转的映射矩阵。

```

CvMat* cv2DRotationMatrix(
    CvPoint2D32f center, //旋转中心
    double angle, //旋转角度
    double scale, //旋转的缩放尺寸, 本文设为1
    CvMat* map_matrix //带入cvWarpAffine中的矩阵
);

```

参数中的 `angle` 取旋转角度 θ 的负值。为了防止出现图像旋转后超出 256×288 的全景框, 本文选取前景区的几何重心为旋转中心为 O 。旋转完成后, 以中心 O 与全景框的中心坐标(128, 144)为参考点对进行平移, 以保证最终的指纹对象位于图像正中心位置。

这样, 就完成了对倾斜指纹的旋转处理, 在此基础上继续进行未完成的预处理及特征提取。指纹图像校正的算法流程图如下:

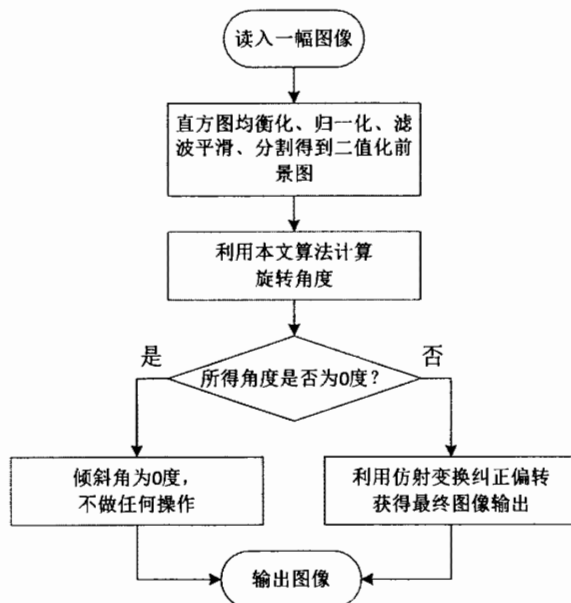


图4-13 旋转校正流程图

4.4.5 实验结果及分析

图 4-14 是部分效果图, 依次是三种方法的结果。第一组中 (a) 并未有效校

正，通过（b）图的弥补，（c）图已接近摆正；第二组结果则正好相反，图（a）纠正了图（b）的错误；最后第三组则正好将前两幅图的误差相互抵消，最后获得理想的输出结果。



图4-14 部分旋转校正效果图

本文实验用的指纹图像来自FPC1011F指纹传感器采集建立的指纹库，从库中共抽取300幅图像，分别来自30枚不同的手指。为了提高测试覆盖率，本文选择了100幅未发生旋转的指纹图像，100幅是指纹向左旋转，另100幅是指纹向右旋转，旋转角度分别为22.5°、45°、67.5°和90°。同时，由于采集头的采集面积较小，无法完整包含整个指纹的轮廓形状，故实验样本中也存在一些人工也无法判断是否旋转的图像，我们可以认为不需要校正。

分别实现了文献[43]和文献[44]的方法后，与本文的算法对比的最终结果如表4-1。

表4-1 指纹旋转校正结果

| 类别 | 校正正确 (%) | 过校正 (%) | 欠校正 (%) |
|----------|----------|---------|---------|
| 文献[43]方法 | 94.7% | 5.3% | 0% |
| 文献[44]方法 | 90.7% | 0.3% | 9% |
| 本文方法 | 99.33% | 0.67% | 0% |

实验结果表明,该算法吸取了前人方法的优点,能够保证对采集时偏转不大的指纹图像不进行旋转,对偏转较大的指纹图像准确检测到旋转角度,并进行校正,避免了文献中出现的错误校正现象。同时,较之文献[46],可以最大限度保证方向场正确,提高后续提取奇异点的精确度;与文献[47]相比,仅需处理灰度图像,无需提取细节特征,算法易于实现。但实验结果也表明,仍然存在两幅过校正的指纹图像,针对这种情况,我们可以引入反馈机制。按照本文的处理流程,指纹图像在计算方向场之前先要通过校验保证图像摆正,如果可以利用类似文献[41]中的方法,借助方向场找出校正得不对的图像,并计算偏差的角度,就能够进行有反馈的处理,根据补偿的角度重新校正。

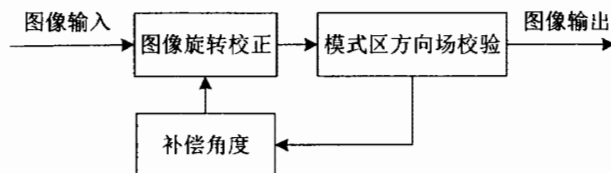


图4-15 反馈处理机制

4.5 奇异点提取

奇异点是指纹纹理结构中心位置的一种描述,也是指纹分类最主要的类型特征之一。提取奇异点主要有以下两个作用:

1. 作为分类的依据:奇异点的数目和分布描述了指纹的全局特性,这种特性经常被用来作为粗分类的标准。
2. 作为定位的基准。在指纹识别过程中,为了提高处理速度,常常以中心点作为两幅图像配准的基准点。

综上,奇异点提取直接影响到指纹分类和指纹匹配的准确率和速度。其中,中心点存在于大部分的指纹图像当中,但是三角点由于指纹形状、采集面积等的限制不具有普遍性,所以本文主要提取中心点。

4.5.1 传统的 Poincare index 方法

目前,基于 Poincare index 的方法是指纹中心点提取方法中的主流,其思想是依据指纹图像的方向旋转特性区分奇异点和普通点:在当前点的邻域内,围绕该点的闭合曲线顺时针旋转一周,统计方向的变化量之和,针对于不同属性的点,这种旋转变化的规律也是不同的。中心点的方向变化量为 -180° ,三角点的方向变化量为 180° ,普通点的方向变化量为 0° [44]。

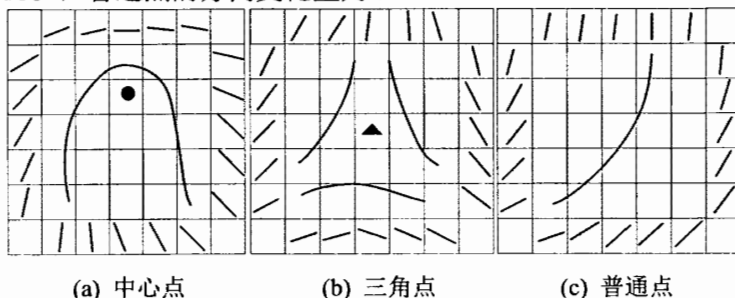


图4-16 中心点、三角点及普通点周围方向规律示意图

根据前文，我们求得的方向场在 $[0 \sim \pi]$ 上，点 (i, j) 处的 Poincare index 值 $P(i, j)$ 可按式(4.3)计算（假设该闭合曲线上有 m 个元素）：

$$Poincare(i, j) = \frac{1}{2\pi} \sum_{k=0, \dots, m-1} \Delta(k) \quad (4.3)$$

$$\Delta(k) = \begin{cases} \delta(k) & |\delta(k)| < \frac{\pi}{2} \\ \pi + \delta(k) & \delta(k) \leq -\frac{\pi}{2} \\ \pi - \delta(k) & otherwise \end{cases} \quad (4.4)$$

$$\delta(k) = D_k - D_{(k+1) \bmod m} \quad (4.5)$$

其中， D_k 是以给定点为中心的封闭曲线上第 k 个点的方向， \bmod 表示求模运算。如果 Poincare Index 值为 $\frac{1}{2}$ ，那么点 (i, j) 就被视为中心点，如果 Poincare Index 值为 $-\frac{1}{2}$ ，那么点 (i, j) 就被视为三角点。

传统的 Poincare Index 方法具有很多缺点，如在计算过程中需要遍历图像中所有的像素点，这大大降低了检测效率；计算索引时只用一条封闭曲线，提取出大量的伪中心点；计算一个点周围的方向改变累积量时，求得的 Index 值并不是准确地等于 $\frac{1}{2}$ ，而是在 $\frac{1}{2}$ 附近的一个很小的邻域内： $[-\frac{1}{2} - \delta \sim \frac{1}{2} + \delta]$ ；有些噪声点的方向变化累积量正好满足条件，但并不符合中心点的规律等，这些因素都导致传统的 Poincare Index 方法抗噪性变差。针对这些问题本文提出了求取指纹图像中心点的改进方法。

4.5.2 本文采用的提取方法

本文在前人成果的基础上提出了一整套提取中心点并去伪点的方法。提取过程的流程图如图 4-17：

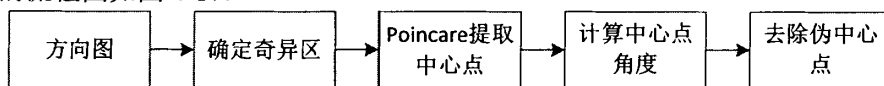
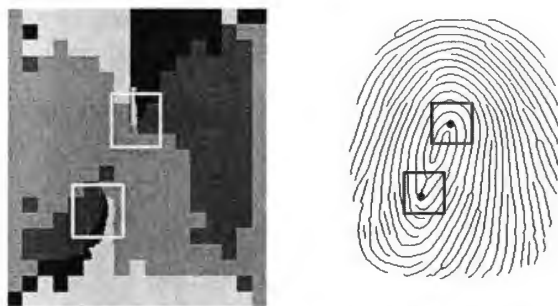


图4-17 提取中心点流程

4.5.2.1 确定中心点所在的区域

根据 3.6.2 节对方向场统计的规律，中心点附近区域的方向变化十分剧烈，重新计算点方向的区域就是中心点候选区域。因此，为了验证粗定位的准确性，采用文献[48]中归域化的方法，将方向场归域化为 $(\Omega_{(0, \pi/4)}, \Omega_{(\pi/4, \pi/2)}, \Omega_{(\pi/2, 3\pi/4)}, \Omega_{(3\pi/4, \pi)})$ 四个方向范围，以图 4-18 中的指纹图像为例，矩形框中四个方向范围是基本均匀分布的，说明其中不同的方向值较多，中心点有可能在该范围内。根据以上分析，同时为了提高算法搜索速率，以中心点候选区每个像素点为中心，统计其 60×60 大小范围内的方向分布，通过判断每个图像块是否符合归域化规律来确定中心点候选集。

实验过程中观察图 4-19 的方向图可以发现，前景与背景的交接处方向变化也比较剧烈，可能会影响判断，出现伪中心点。针对这种情形需要进一步利用后续的方法去伪中心点。



(a) 指纹图像方向场灰化图 (b) 细化图提取中心点示意图

图4-18 指纹方向灰化图与细化图的中心点位置对照图



(a) 指纹图像方向场灰化图 (b) 细化图提取中心点示意图

图4-19 存在伪中心点时的对照图

4.5.2.2 改进的 Poincare 方法提取中心点

通过改进传统 Poincare 方法，提取具体步骤如下：

1. 根据中心点基本分布在前景区的中心附近这一规律，首先判断当前点的 9×9 区域内 80 个像素点是否存在分布在背景区的点，有则继续判断中心点候选区的下一个像素点。

2. 为了克服传统 Poincare index 方法提取不准确的缺点，采取两条长度分别为 3×3 和 5×5 的封闭曲线顺时针计算每个点的索引值，如图 4-20。只有当两次的 Poincare index 索引值均满足式(4.6)，并且符合文献[49]中的规律时，该点才是粗提取的中心点。

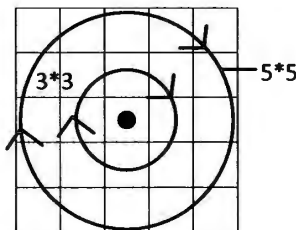


图4-20 3×3 和 5×5 的封闭曲线

$$|Poincare(i, j) + \pi| < 0.01 * \pi \quad (4.6)$$

3. 完成步骤 1 和步骤 2 后, 我们可以得到一个候选中心点集, 其中可能包含相距仅一个像素点的冗余点, 如图 4-21。根据放大的图像, 获得中心点 A 的唯一坐标是(97, 107), 而中心点 B 是由四个候选点组成的, 坐标分别为(126, 178)、(127, 178)、(126, 177)、(127, 177), 即真正的中心点候选区附近可能会有多个中心点满足条件, 这是中心点区形成的聚类, 但是我们只需要一个特征最明显的点, 所以每得到一个粗提取的中心点, 就需要计算其与已存储的所有中心点之间的欧氏距离, 当距离小于两个纹线距离 (15 个像素) 时, 就将这两个点的坐标平均值存入中心点集。最终在无边界干扰的情况下得到两个相对准确的中心点, 坐标分别为(97, 107)、(126, 188)。



图4-21 中心点聚集情况的示意图

4.5.2.3 计算中心点角度

在提取中心点的同时, 为了后续的指纹分类工作, 还要求解中心点附近纹线的走向, 我们将其称为拱形方向, 也就是中心点角度。观察图 4-22 发现中心点附近往往呈现“U”形或者“∩”形方向趋势特征^[15], 尽管由于图像扭曲的影响, 不同指纹图像可能表现出一些差异, 但总体上仍会保持这种特性。“U”形的出口方向即为中心点的方向, 也就是中心点附近的对称轴方向。

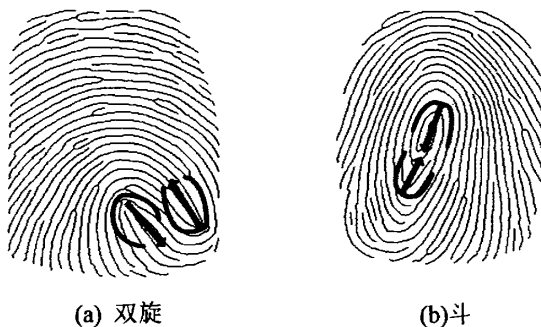


图4-22 中心点附近“U”形和“∩”形示意图

在前面的工作中, 已经将倾斜的指纹图像恢复到手指摆正的状态, 那么针对存在中心点的指纹, 只要获得了指纹的出口方向就可以确定指纹的类别, 具体求解方法如下:

1. 以图 4-23 为例, 对 2.2.2 节候选中心点集中的每一个点, 我们从中心点开始, 以一个纹线间距 7 个像素步长沿拱形出口反方向扫描 3 个像素点, 如图 4-23 中绿色箭头所示, 若满足条件 1, 继续判断, 否则从点集中剔除此点。



图4-23 中心点方向示意图

2. 实际应用中会遇到“U”形严重倾斜的情况，但是其切线方向上的点仍然满足条件2的规律，所以在得到了中心点角度后，以中心点为原点，切线方向为作 x 轴，法线方向为 y 轴建立笛卡尔坐标系，如图 4-23 中黄色箭头所示，计算中心点的二元一次切线方程，遍历这条线上近似像素点的方向，若满足角度一对一对称的条件，利用步骤3计算最终角度，否则删除该点。

3. 继续以 7 个像素步长向拱形出口方向扫描 N 个像素点的方向场角度，如图 4-23 中蓝色箭头线所示，若满足条件 3，则根据公式(4.7)统计 N 个角度与竖直方向的平均偏差。其中，以 7 个像素为步长扫描是为了避免噪声污染并提高扫描速度。最终利用公式(4.8)获得中心点角度方向 θ_{axis} ，方向示意图如。

$$\Delta\theta_{\cup\cap} = \left(\sum_{i=1}^N \left(\theta_{\cup\cap} - \frac{\pi}{2} \right) \right) / N \quad (4.7)$$

$$\theta_{axis} = \Delta\theta_{\cup\cap} + \frac{\pi}{2} \quad (4.8)$$

条件 1：以中心点为原点 O 做竖直轴，沿“U”形（或者“∩”形）开口的反方向的点集方向是趋于水平的。

条件 2：中心点附近左侧方向角与相应的右侧方向角近似呈现对称分布（由于图像扭曲，有时两侧并非呈现严格的“U”形，允许存在 $\Delta\sigma$ 的偏差），即一对一地满足与水平方向夹角近似相同。

条件 3：若为“U”形，则上方所有像素点角度呈现竖直方向快速发散的趋势，即中心点上方与 $\pi/2$ 夹角为锐角，下方相匹配的点的角度则趋于水平，变化较为缓慢；相反，若为“∩”形，则呈现相反的趋势。

4.5.2.4 去伪中心点

由于噪声、方向场求解不精确等影响，中心点集中仍可能存在为数不多的伪点，而中心点是否能准确提取至关重要，所以在提取中心点并计算角度过程后需要去伪中心点。本文重点将人的一些常识经验总结成规则，根据指纹的纹线变化规律完成去噪工作：

1. 在提取到候选中心点集之后，指纹前景区的边缘区域方向场变化迅速，纹线一致性较差，易存在伪中心点。我们可以借用图像增强处理中使用过的交叉模板的思想，当前点附近 11×11 大小的十字模板中存在背景点，则标记该点是边界伪中心点。

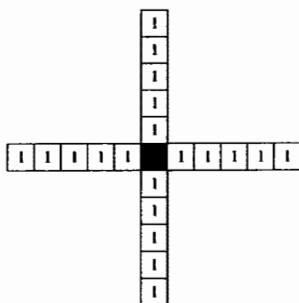


图4-24 11×11十字去噪模板

2. 考虑到中心点附近归域化的特性, 通过判断每个中心点附近 60×60 大小模板范围内的点是否符合平均分布在归域化的四个方向场范围内, 可以去除 1 中未去掉的边界点和噪声点。

3. 利用 4.5.2.3 中求解中心点的方向, 若不满足其中的条件 1、条件 2 和条件 3, 也可以认为是伪中心点并去除。

4. 若中心点集数量为 2, 根据经验, 两个中心点周围的纹线走向形如 “U” 和 “∩”, 如图 4-22 所示。若纹线走向对称, 再检查两个点的方向角度是否近似一致, 若不满足, 则依据中心点附近曲率较大的规律去除曲率相对较小的点。

去噪算法可以有效地去除 Poincare index 方法中由于方向场的误差导致的伪中心点, 方法的流程图如图 4-25。

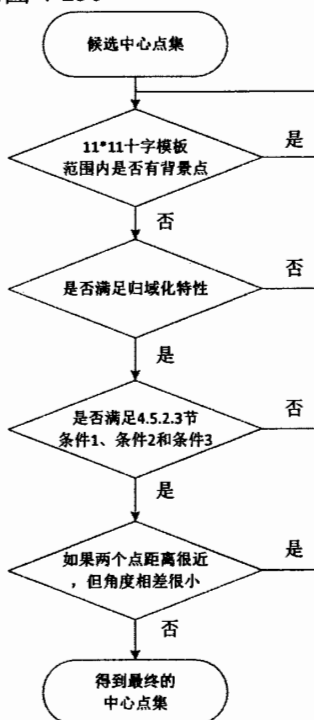


图4-25 中心点去伪算法流程图

4.5.3 实验结果及分析

图 4-26 是部分的比较效果图, 第一组 (b) 图中去除了边界伪点, 第二组 (b)

利用归域化特性去除伪点，第三组（b）根据同一“ \cap ”形不存在两个中心点规律去伪，第四组（b）在提取阶段排除了距离较近的点。

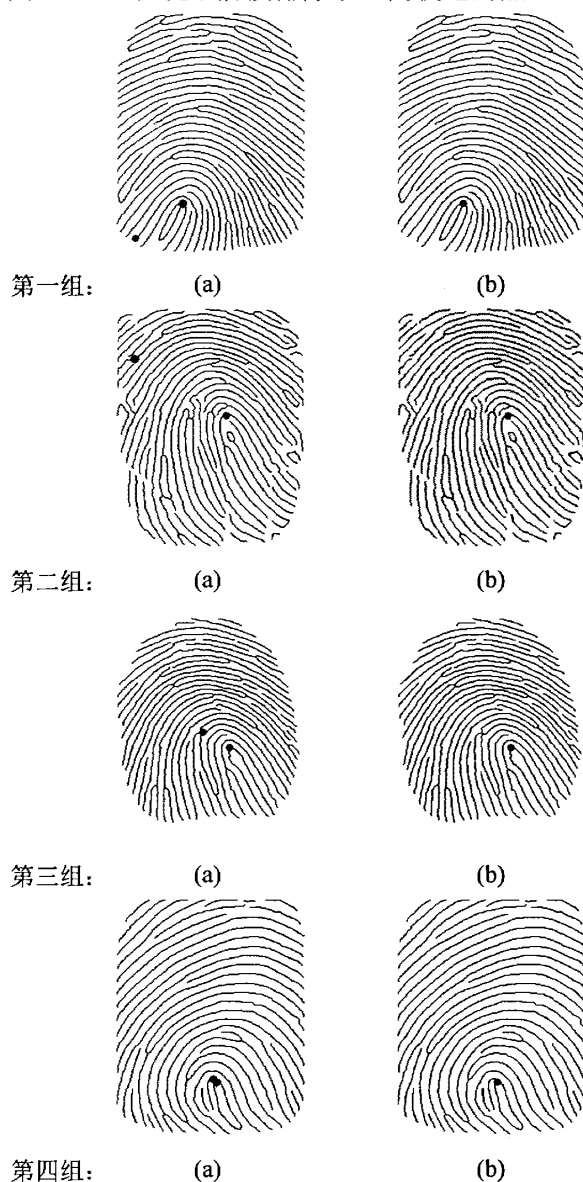


图4-26 中心点提取部分效果图

根据文献[48]，衡量指纹中心点检测定位的主要标准是漏检率（MDR）和误检率（FDR）。本文将指纹采集仪采到的 800 幅的图像作为实验对象，为了模拟实际应用环境，图像中 70%处于旋转状态，40%质量较差。根据 Henry 指纹类型，实验样本中分别包括 200 幅斗型指纹、100 幅双旋型指纹、200 幅右箕型指纹、200 幅左箕型指纹、25 幅有 3 个以上中心点的指纹，为了更好地说明本文算法能够准确判断无中心点的情况，样本中还包括 50 幅弓型指纹、25 幅采集不理想（采集面积较小、图像灰度质量差）的指纹图像。经过人工标记后，实验样本中总共包含 1075 个中心点，本文算法检测到了 1080 个，有 5 幅图像遗漏了共 7 个点，漏检率为 0.65%。另外，有 8 幅图像检测出了 12 个伪点，误检率为 1.12%。通

过表 4-2 可知,与文献[48][49][50]相比,本文算法在控制漏检率和降低错误率、提高定位精度和算法鲁棒性方面都有了很大的改进,若人工干预采集过程,本文的算法可以达到 0%的漏检率和误检率。

表4-2 本文算法统计中心点的结果

| 数据库 | 总数 | 漏检(幅/个) | 误检(幅/个) | 漏检率(%) | 误检率(%) |
|-----|------|---------|---------|--------|--------|
| 图片 | 800 | 5 | 8 | 0.65% | 1.00% |
| 中心点 | 1075 | 7 | 12 | 0.65% | 1.12% |

在实验过程中,将本文算法应用于提取三角点,针对现有的样本中存在三角点的图像,都能够正确地提取,但考虑到三角点出现在采集有效区域的概率较小,所以不再统计实验结果。

与传统的中心点提取方法相比,加入去伪中心点的方法后,本文算法对图像质量不高、方向图求解存在误差等原因产生的伪点有很强的鉴别能力,能够基本满足自动指纹识别系统对中心点提取高精度的要求。

4.6 本文的分类方法

常用的分类方法分为排斥性分类方法(Exclusive Classification)和连续性分类方法(Continuous Classification)^[51],前者是指种类之间相关性较低,类内相关性较高,主要根据指纹全局特征进行分类的方法,而后者是以某一种特征向量为索引,以待查样本为中心,在一定范围内检索,有助于弥补排斥性分类判别过于绝对的不足。

对大规模数据库的自动指纹识别系统来说,高效、准确的分类方法可以帮助提高系统的整体性能,若分类类别不足,则分类的优越性就无法体现出来。所以,本文将在经典的 Henry 分类方法基础上扩大指纹的可分类数目,同时为保证分类正确性,设计了一种基于决策树思想的多级指纹分类方法,分类特征分别包括指纹中心点的数目及附近纹线走向、斗型螺径及箕型中心点附近拱形的开口方向等。

4.6.1 一级分类方法

指纹图像经过旋转校正和中心点提取后,可以根据中心点的数目确定四个类别:第一类——螺旋型,第二类——箕型,第三类——弓型,第四类——杂型,这是一种排斥性分类方法。这里需要说明的是螺旋型表示有两个中心点的情况,包括斗型和螺旋型;箕型表示只有一个中心点的指纹,包括左箕型和右箕型;弓型就是没有中心点的情况;杂型则是指有三个以上中心点的指纹图像。分类结果如图 4-27 所示。

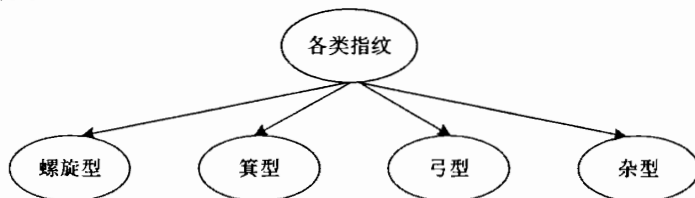


图4-27 一级分类结果示意图

4.6.2 二级分类方法

为了更近一步确定指纹纹型，本文根据得到的中心点纹线方向，对三种纹型进行更详细的分类。

1. 螺旋型可以继续分为斗型和双旋型，图 4-22 给出了两种纹型的特点图，不难发现，中心点的方向基本保持一致，两个中心点附近拱形的开口方向分别是“ \cup ”和“ \cap ”，很多特征都十分接近。文献[52]利用追踪脊线的方法判断，如果从一个中心点沿着一条纹线经过另一个中心点，并且最终回到初始点，则说明是斗型，否则就是双旋型。这种方法虽然非常直观，但在实际应用中跟踪纹线需要做很多的计算及判断，并且在指纹存在较多断裂和模糊的情况下无法准确追踪。所以，本文采用基于点的相对位置方法区分两种纹型。从图 4-28 我们知道，双旋型指纹图像中两个中心点水平间距 X 明显大于其竖直方向间距 Y ，而斗型中心点水平间距要小于竖直方向间距，利用 X/Y 是否大于 1 区分两种纹型。

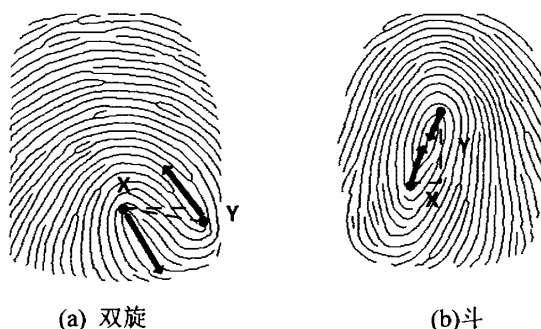


图4-28 双旋型和斗型区别图示

2. 簸箕型纹通常只有一个中心点，且中心点方向与竖直对称轴存在一定的夹角，比如左簸箕纹型相对于竖直方向顺时针旋转，右簸箕纹型相对于竖直方向逆时针旋转。



图4-29 左簸箕和右簸箕中心点方向示意图

弓型和杂型指纹的出现概率一般不高，子样本数量较少，所以无需继续分类。图 4-30 就是二级分类的示意图。

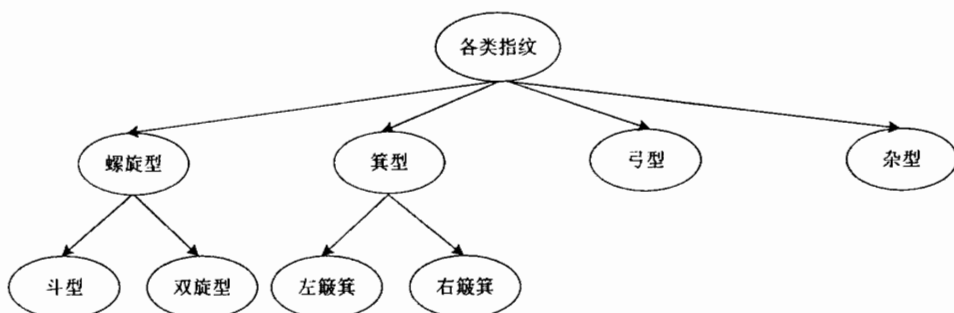


图4-30 二级分类示意图

4.6.3 三级分类方法

在之前的介绍中曾提到，基于 Henry 分类法的结果是不均匀的，这意味着，经过本文的两级分类后，系统穿透系数仍然无法达到大规模指纹匹配的要求，主要在于斗型、左簸箕型和右簸箕型占有较大比例，指纹匹配时仍需要比对一定数量的样本。所以还需要继续细分这三种纹型。

4.6.3.1 针对斗型指纹三级分类

斗型指纹是圆圈型的指纹，通常是 N 个同心圆组成，两个中心点就在圆圈曲率最大的两个位置上。这种指纹的形状与螺丝十分相似，我们可以定义两个中心点之间的距离为螺径，图 4-31 就是螺径的示意图。在中心点精确定位的前提下，同一斗型指纹的螺径是一个近似不变的特征值，也就是说即使对指纹进行高扭曲度的形变，螺径仍在一定弹性范围内是不变的，因此螺径是一种可分类的特征。



图4-31 不同螺径的斗型指纹

本文所研究的指纹图像统一是 256×288 像素大小，所以一般螺径的长度不会超过 150，一般都集中在 20~100 之间。在实际应用中，针对同一枚手指的不同形态，采集到的中心点相对位置会有一定的偏差，为了减小误差带来的影响，我们采用螺径的平方值作为分类的特征，并且采用连续分类策略：首先将 0~14000 的范围平均分为 15 类，即螺径长度为 0~500 为第一类，500~1000 为第二类，其他按 n 到 $n+1000$ 的分为一类，若螺径大于 14000 则统一归为一类，总共分为 16 类。

就指纹仪采集到的 224 幅斗型指纹图像统计螺径的统计结果如表 4-3。

表4-3 螺径大小统计

| 螺径 | 指纹数 | 所占比例 | 螺径 | 指纹数 | 所占比例 |
|-------------|-----|-------|-------------|-----|-------|
| 0~500 | 8 | 3.57% | 500~1000 | 54 | 24.1% |
| 1000~2000 | 28 | 12.5% | 2000~3000 | 56 | 25% |
| 3000~4000 | 5 | 2.23% | 4000~5000 | 5 | 2.23% |
| 5000~6000 | 21 | 9.38% | 6000~7000 | 16 | 7.14% |
| 7000~8000 | 3 | 1.34% | 8000~9000 | 8 | 3.57% |
| 9000~10000 | 3 | 1.34% | 10000~11000 | 5 | 2.23% |
| 11000~12000 | 4 | 1.79% | 12000~13000 | 2 | 0.89% |
| 13000~14000 | 6 | 2.69% | 14000~ | 0 | 0% |

即使连续分类在一定程度上提高了正确率, 由于同一枚指纹采集环境各异, 经过一系列预处理后所得到的螺径大小也必定存在一定的误差, 所以当指纹无法与当前螺径范围中的样本正确比对时, 我们需要扩大搜索半径, 降低拒识率。

当输入指纹图像时, 定义为 O , 第一次索引的类别为 T , 在数据库中搜索与 O 相同的指纹的方法如下:

1. 计算螺径, 并且设置搜索半径初始值 $k=0$;
2. 将第 T 类中的指纹与 O 图比对, 比对成功则退出, 否则进行步骤 3;
3. 将第 $T-k$ 类中的指纹与 O 图比对, 比对成功则退出, 否则进行步骤 4;
4. 将第 $T+k$ 类中的指纹与 O 图比对, 比对成功则退出, 否则进行步骤 5;
5. 增加比对半径 k 的值, 重复步骤 3 和步骤 4, 直至比对成功或者 $k>2$ 为止。

这一整个过程中如何确定搜索半径 k 是关键, 为了获得正确的 k , 我们按照上述方法与样本数据库中的指纹进行匹配, 在不同的搜索半径下统计匹配结果的拒识率, 当拒识率为 0% 时即可获得最大搜索半径。但搜索半径过大也会降低搜索速度, 所以要综合经验选取合适的 k 。

4.6.3.2 针对簸箕型指纹三级分类

簸箕型指纹只具有一个中心点, 不具有斗型的螺径特征, 但簸箕型指纹中心点的拱形方向是人工分类方法中一个常用的特征。图 4-32 中两个来自不同手指的指纹的拱形方向明显不同。一般的方法中并未对指纹图像进行旋转校正, 所以常常忽略这一特征, 但本文已将指纹的对称轴旋转至竖直方向, 所以针对簸箕型指纹的三级分类特征定义为指纹中心的拱形方向, 该指标与斗型的螺径一样也是一个连续性指标。

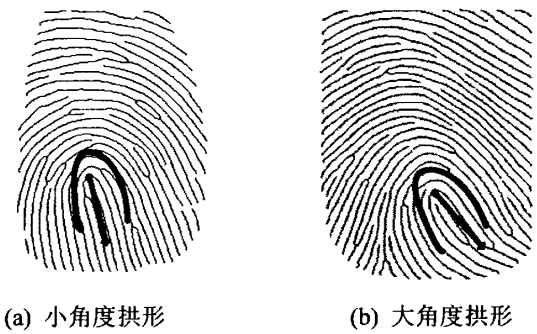


图4-32 不同拱形的出口方向示意图

因为本文所计算的方向都是采用弧度制，范围在 $[0,\pi]$ 内，所以拱形方向与竖直对称轴的夹角绝对值范围是 $[0,\pi/2]$ 。通常，方向差值分布在 $[0,\pi/4]$ ，我们将 $[0,0.7]$ 范围平均分为7类，即 n 到 $n+0.1$ 是一类，将大于1.5的定义为第八类。具体的搜索方法与斗型三级分类的方法相似。

无论是左簸箕或是右簸箕，其拱形方向与 y 轴的夹角是在相同范围内的，所以，本文以420幅不分左右的簸箕型指纹图像作为统计对象，其夹角绝对值的分布如表4-4。

表4-4 簸箕夹角方向统计

| 夹角绝对值 | 指纹数 | 所占比例 | 夹角绝对值 | 指纹数 | 所占比例 |
|---------|-----|--------|---------|-----|--------|
| 0~0.1 | 31 | 7.38% | 0.1~0.2 | 54 | 12.86% |
| 0.2~0.3 | 96 | 22.86% | 0.3~0.4 | 92 | 21.9% |
| 0.4~0.5 | 71 | 16.9% | 0.5~0.6 | 38 | 9.05% |
| 0.6~0.7 | 17 | 4.05% | 0.7~ | 21 | 5% |

4.7 实验结果及分析

利用以上的三级分类方法，可以得到指纹分类的决策树示意图4-33。

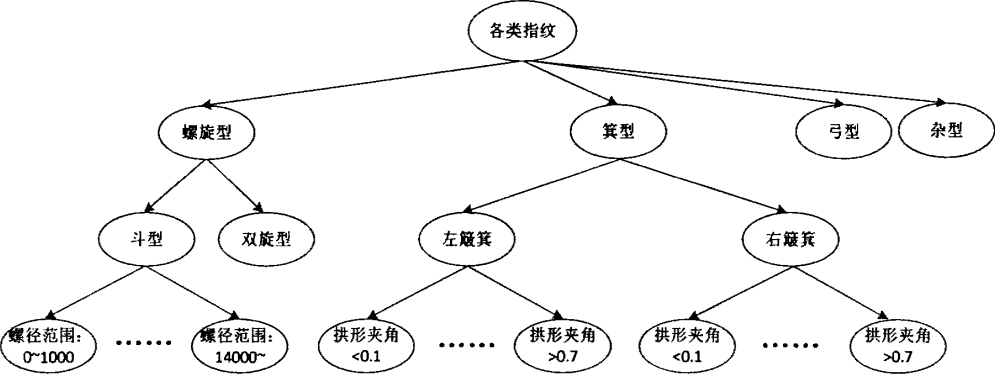


图4-33 三级分类结果示意图

本文直接对已提取特征点的 800 幅指纹图像进行分类实验,并且已经剔除了有效面积不足、中心点偏移等质量不合格的指纹图像。根据前文分析,一、二级分类是排斥性分类方法,为了清晰地说明其分类的准确率,我们使用混合矩阵评估实验结果,分类效果如表 4-5 所示。

表4-5 指纹分类混合矩阵

| 实验测得 分类结果 | 人工分类结果 | | | | |
|--------------|--------|-----|-----|-------|-----|
| | 斗型 | 双旋型 | 左簸箕 | 右簸箕 | 其他 |
| 斗型 | 190 | 4 | 1 | 0 | 0 |
| 双旋型 | 6 | 94 | 0 | 1 | 0 |
| 左簸箕 | 1 | 1 | 196 | 3 | 2 |
| 右簸箕 | 1 | 0 | 2 | 195 | 1 |
| 其他 | 2 | 1 | 1 | 1 | 97 |
| 总数 | 200 | 100 | 200 | 200 | 100 |
| 分类正确率 | 95% | 94% | 98% | 97.5% | 97% |

根据实验结果统计,本文分类算法的平均正确率是 96.5%,与文献[52]和文献[53]相比,具有明显的优势,究其原因,主要得益于中心点提取准确率较高。

第三级分类属于连续性分类,其分类准确率受最大搜索半径的影响,在不考虑实时性要求的情况下,分类准确率是 100%,所以,系统总的分类正确率就是 96.5%。

4.8 本章小结

本章主要介绍了基于大规模样本数据库的指纹分类算法。在分类之前,首先对旋转的指纹图像进行校正、然后提取分类特征,最后根据不同的特征完成多级分类。

校正旋转指纹图像的算法主要包括旋转角度的计算和基于角度进行仿射变换的过程。首先,求取旋转角度,根据图形轮廓分别计算中位线的角度和最小外接椭圆的偏转方向,按照一定权重计算最终的纠正偏角,弥补了两者欠校正和过校正的缺点。然后,使用数字图像处理中常用的仿射变换对指纹图像进行旋转校正得到最终的输出图像。

特征提取主要完成中心点的数量、绝对方向的提取、斗型指纹螺径的计算、以及箕型指纹中心拱形附近的纹线走向的确定。在中心点的提取上,本章对传统的Poincare算法进行了改进,并提出了智能化的去伪中心点的方法,提高了特征提取的准确率,为后续的分类操作提供了支持。

最后,提出了一种三级指纹分类机制,第一级分类根据中心点数目将指纹分为四类;第二级分类针对螺旋型中双旋型和斗型指纹中心点分布的区别,将螺旋型继续细分,同时,利用箕型中心拱形开口方向不同又增加了左簸箕型和右簸箕型;第三级分类利用螺径和拱形纹线方向又将斗型、左簸箕型、右簸箕型分为16类和8类。

第五章 基于点模式的指纹特征匹配

无论是基于大规模指纹库的指纹检索系统还是普通的嵌入式指纹锁,最后都需要准确性高,鲁棒性好的指纹匹配模块,它是自动指纹识别系统的核心。指纹匹配的最终目的是判断两幅指纹图像是否来自同一枚手指,主要通过将输入的指纹图像与指纹库中的样本进行比对,计算匹配的相似程度,最终给出结果。本章的匹配算法并不明确区分比对的是整个指纹库还是分类后的子样本,因为在两种应用场景下匹配的核心方法是相同的。

5.1 指纹特征匹配概述

目前,指纹的匹配算法根据其所使用的匹配特征大致分为:基于点模式^[54]的匹配算法、基于纹线^[55]的匹配算法和基于纹理结构^[56]的匹配算法。其中基于点模式的匹配算法因其算法简单高效得到了广泛的关注;基于纹线的方法一般在指纹局部细节特征点较少,点模式方法无法发挥作用时使用,但纹线特征的提取和匹配需要大量的计算,并且对细节点提取的准确性要求较高;基于纹理模式的方法能有效提高识别性能,但代价是所需的时间和空间远大于前两种。

基于点模式的匹配算法一般是利用细化后提取的细节点进行匹配。Jain 等^[54]提出一种基于几何关系的细节点校准匹配方法,主要是基于极坐标具有平移、旋转不变性的思想,对细节点进行配准匹配,虽然比较耗时,但效果良好。在该方法中最关键的就是准确找出参考点对,由于指纹图像中存在形变并且检测到的特征带有一定的误差,导致很难准确地获取对应参考点^[57]。迄今为止,很多文献对此问题做了研究,杨栋等^[57]利用细节点周围三近邻的细节点建立特征向量,利用三个点与中心点的角度差和距离差筛选参考点对。Janm^[58]提出了一种基于二叉树的线性时间匹配算法,先是提取出指纹图像的中心点,然后以此为基准向邻域扩散,寻找合适的参考点对。尹义龙等^[59]采用三角网络的方法寻找目标点对,虽然准确率较高,但是需要匹配的三角形较多,耗费大量的存储空间。Nandakumar^[60]使用局部细节点相位谱来对给定细节点邻域的局部结构进行编码,根据两枚指纹的细节点相位谱间的汉明距离帮助定位参考点对。

本文采用基于点模式的匹配算法,为了提高运行速度,分为两级匹配。首先,初匹配阶段参考三角形匹配的方法,但细节点之间的局部结构的区分性具有一定的局限性,抗噪声、形变影响的能力较弱,在匹配之前先验证细节点附近的方向场和相对位置,大大降低了伪细节点被匹配的概率。然后,根据初匹配的结果,结合文献^[54]参考点配准的思想和文献^[61]可变大小的限界盒方法完成全局匹配。

5.2 细节点提取

基于点模式的指纹识别过程中,主要比对的是局部特征,而依据前文的分析,端点和分叉点出现概率最高,是主要的识别特征。

目前,指纹细节点提取方法主要分为三种:基于灰度图像、基于二值图像和基于细化图像的提取方法^[21]。比较而言,第一种方法因为避开了正常的预处理

过程,所以计算非常复杂,可实现性较低;第二种方法的缺陷在于二值图像会产生很多伪细节点,鲁棒性也较差。鉴于本文对指纹图像已进行了相对全面的预处理操作,所以直接采用基于细化图像的提取细节点方法。在细化图像中,细节点的类型与邻域 8 个点的灰度值之间是符合一定的规律的,如图 5-1 所示。

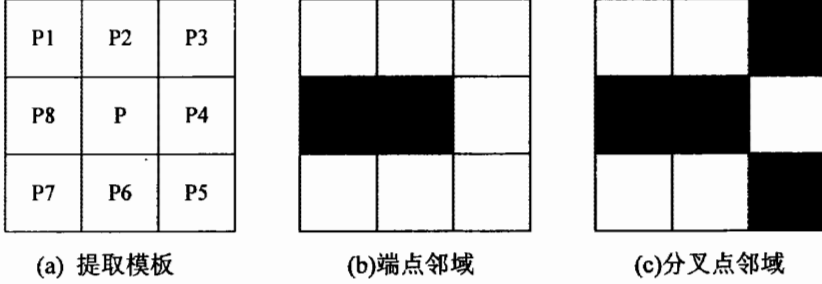


图5-1 指纹纹线端点和分叉点的 8 邻域图

设 P 为目标点, P_1, P_2, \dots, P_8 依次是 P 点的 8 邻域像素点,沿顺时针方向排列, $R(1), R(2), \dots, R(8)$ 分别是点 P_1, P_2, \dots, P_8 的灰度值。将这 8 个点进行循环比较,若 0,255 变换两次,并且 8 邻域像素点的灰度值之和为 $7*255$,则待测点为端点;若 0,255 变换了六次,并且 8 邻域的灰度值之和为 $5*255$,则待测点为分叉点^[29]。具体计算方法如公式(5.1)~(5.4)所示:

如果 P 为端点,则 8 邻域满足:

$$C_N = \sum_{i=1}^8 |R(i+1) - R(i)| = 2 * 255 \quad (5.1)$$

$$F_N = \sum_{i=1}^8 |R(i+1) + R(i)| = 7 * 255 \quad (5.2)$$

如果 P 为分叉点,则 8 邻域满足:

$$C_N = \sum_{i=1}^8 |R(i+1) - R(i)| = 6 * 255 \quad (5.3)$$

$$F_N = \sum_{i=1}^8 |R(i+1) + R(i)| = 5 * 255 \quad (5.4)$$

5.3 细节点去伪

基于细化图像的特征提取方法虽然原理通俗易懂,计算速度快,但细化处理本身会改变部分原始图像的特征,导致提取出许多错误的细节点。另外,根据前文指纹图像增强部分内容可知,指纹断裂和粘连会形成伪端点和伪分叉点,虽然通过预处理后部分噪声已经被消除,但并不能 100%地达到要求。因此,清除伪特征是十分有必要的。具体的去伪算法描述如下:

1. 若当前细节点不是背景点,则检查该点是否靠近边缘,若靠近,则删除,重新判断下一个细节点,否则继续进行步骤 2。这一操作可以有效地消除边缘效应,由于指纹采集器自身原因而造成指纹图像被截断产生伪特征,设定一个阈值,即可去除不可靠的点。

2. 检查当前细节点周围的方向场变化是否剧烈,因为端点和分叉点附近的方向变化相对于中心点来说平缓一些,所以若变化剧烈,则可能是伪细节点。

3. 指纹中的纹线一般不会太短，可以两两端点循环计算欧氏距离，当距离小于一定阈值时，继续判断两个端点的方向角是否相近，是则认定是纹线断裂造成的临近伪细节点。

4. 当两个分叉点距离较近时，很可能是小桥或者误连，所以只要追踪其中一个分叉点相连的三条纹线，若在一定步长内检测到另一个分叉点，并且两点的方向大致相反，就可以判定为伪分叉点，将两点都删除。

5. 一个完整的指纹，其细节点数目过多或者过少都是不正常的，所以当细节点较少时可以提示重新采集指纹，当较多时则返回步骤 1，调整比对阈值和搜索步长，重新进行判断。

经过去除伪特征后的结果如图 5-2:



(a) 未经去伪算法处理的结果图 (b) 经过去伪算法处理后结果图

图5-2 基于细化图的特征提取方法去伪前和去伪后的结果对比图

5.4 特征向量的表示

基于点模式匹配的方法归根结底是将计算两个细节点的相似度转换成对应特征向量的相似度。所以对每个细节点要建立特征向量，其方法如下：

首先建立每个细节点的基本特征向量，在指纹图像上以左下角顶点为坐标原点，水平方向为横坐标，垂直方向为纵坐标建立笛卡尔直角坐标系。为每个细节点记录下细节点类型 Type，坐标位置(x, y)，方向角度 θ 。在计算 θ 时，并不能简单定义为该点在方向场中的值，这是因为本文采取的是块方向场方法，其结果并不精确。为了减少误差，本文以每一个细节点为起点，细节点若为端点，则沿与其连接的脊线进行跟踪，并将终点和起点相连直线的倾斜角作为细节点方向；细节点若为分叉点，则分别沿与其相连的三条脊线进行跟踪，然后计算三条线的夹角，以最小夹角所对的那条脊线方向作为分叉点的方向，不同于方向场的是，细节点的方向角度范围是 $[0, 2\pi]$ 。

此外，还需要为初匹配建立表征细节点附近局部子结构的特征向量。虽然指纹图像会存在一定的平移、旋转或者非线性形变，但细节点相邻的纹线结构以及相邻细节点间的欧式距离、相对角度和穿过的纹线数目不会发生太大的改变。如图 5-3 所示，两幅图像来自同源指纹，其细节点附近的方向场分布规律及邻近细节点相对位置基本相同。



图5-3 同源指纹的细节点分布关系

因此，可以利用指纹的一些局部结构信息建立特征向量，帮助确定配对基准点。具体步骤如下：

1. 分别位于两枚相同指纹中的两点，若为匹配的点对，则该点周围方向场的分布必然相同或近似。受基于纹线匹配中的离散点采样方法的启发，以一个细节点为中心 O ，以给定半径 R 做圆，在圆上找到三个点 A 、 B 、 C 恰好构成等边三角形，记录三角形三个顶点的方向场 θ_A ， θ_B ， θ_C 。该特征的匹配相对于即将介绍的邻近三角形要苛刻得多，所以可以有效降低误识的概率，其结构模型如图 5-4。

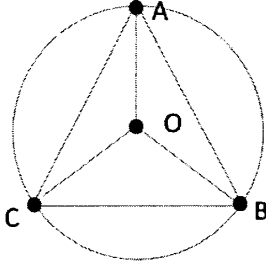


图5-4 局部全等三角形结构模型

2. 步骤 1 是统计了每个细节点附近的方向场、频率场分布情况，下面接着记录不同细节点之间的相对位置和方向等。对每个细节点，取与该中心点距离大于 R (R 取一个纹线周期) 的最近两个点 A 、 B 共同组成三角形，如图 5-5， $A(x_A, y_A)$ ， $O(x_O, y_O)$ ， $B(x_B, y_B)$ 之间的距离按大小顺序分别为 D_L ， D_M ， D_S ，长边 D_L 对应的点为 O_L ，中长边 D_M 对应 O_M ，短边 D_S 对应点 O_S 。距离计算方法如公式 5.5：

$$D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (5.5)$$

在特征向量中还需要记录三个点之间的方向角度关系： α_L ， β_{LM} 和 β_{LS} ， α_L 是长边所对应的角度， β_{LM} 和 β_{LS} 是长边的两个端点和另一个端点的方向之差。为了使角度更具区分性，采用文献[57]的方法，将角度变化的顺时针或逆时针方向考虑进去。计算公式如 5.6 和 5.7：

$$\Delta\beta = \theta_{\text{dest}} - \theta_{\text{src}} \quad (5.6)$$

$$\beta = \begin{cases} -\Delta\beta - 2\pi & -2\pi \leq \Delta\beta < -\pi \\ -\Delta\beta & -\pi \leq \Delta\beta < \pi \\ 2\pi - \Delta\beta & \pi \leq \Delta\beta < 2\pi \end{cases} \quad (5.7)$$

θ_{src} 代表源点的方向角度, θ_{dest} 代表目标点的方向角度, 从源点到目标点需要旋转 β , 顺时针为正值, 逆时针为负值。

尽管欧式距离已经可以反映细节点之间的距离关系, 但是由于非线性形变带来的纹线间疏密的变化会影响匹配时的准确度, 所以, 本文另外存储了三个顶点之间的纹线频率: F_{LM} , F_{MS} , F_{LS} 。纹线频率就是点与点之间连线所穿越的脊线数, 可以利用 Bresenham 画线算法计算, 其基本原理就是连接起点和终点, 计算该直线与事先构建的虚拟网格的交点, 然后统计与交点最近的黑色像素点的总个数^[61]。

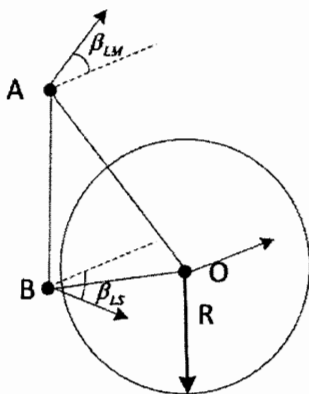


图5-5 局部三角形结构模型

综上, 特征向量的表示如图 5-6:

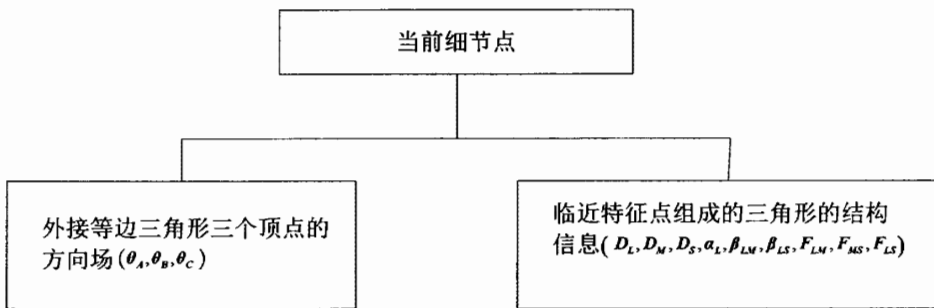


图5-6 特征向量存储模型

5.5 初匹配

首先, 定义指纹匹配模块输入的两个关键的点集: P 和 Q 。其中 P 是新采集进来的待识别图像中提取出来的细节点集, 点集 Q 是预先存储在数据库中的指纹图像中提取出来的细节点集。设点集 P 的大小为 M , 点集 Q 的大小为 N , 把这两个细节点集分别表示为:

$$P\{(P_1, P_2, \dots, P_m)\}$$

$$Q\{(Q_1, Q_2, \dots, Q_m)\}$$

(P_1, P_2, \dots, P_m) 和 (Q_1, Q_2, \dots, Q_m) 分别是两幅图像的特征向量，经过比对后，若两幅图像的特征向量完全匹配，则点集 P 经过某种变换（旋转、平移）就可以得到点集 Q 。

本文参考了文献[23]中的矢量三角形方法，结合细节点外接全等三角形的定义，对指纹进行初匹配，目的有三个：一是对于前景面积较大，特征明显的指纹图像，通过简单的局部细节匹配就可以得出结果，避免计算量较大的全局匹配处理；二是传统的点模式方法中，往往是穷举所有参考点对，最后实现最优匹配，初匹配可以剔除部分不满足条件的比对样本图像，并挑选出最有可能的参考点对，降低全局匹配的时间；三是将初匹配的结果作为一项权重，加入到最后的匹配评分中。具体步骤如下：

- 1) 在样本 Q 中查找与 P 中的点 P_i 类型一致的点 Q_i ：

$$P_i.Type = Q_i.Type \quad (5.8)$$

- 2) 比对两个点的外接全等三角形的方向场：

$$|(P_1.\theta_A - P_1.\theta_O - Q_1.\theta_A - Q_1.\theta_O)| < \varepsilon \quad (5.9)$$

$$|(P_1.\theta_A - P_1.\theta_B - Q_1.\theta_A - Q_1.\theta_B)| < \varepsilon \quad (5.10)$$

$$|(P_1.\theta_B - P_1.\theta_C - Q_1.\theta_B - Q_1.\theta_C)| < \varepsilon \quad (5.11)$$

$$|(P_1.\theta_A - P_1.\theta_C - Q_1.\theta_A - Q_1.\theta_C)| < \varepsilon \quad (5.12)$$

- 3) 接着判断两个点多在的三角形是否近似全等，先是要查看三角形的边长 D_M ， D_S 大小是否分别相同，考虑到非线性形变带来的误差，本文设定一个阈值 ε ，若满足：

$$|P_1.D_M - Q_1.D_M| \leq \varepsilon \quad (5.13)$$

$$|P_1.D_S - Q_1.D_S| \leq \varepsilon \quad (5.14)$$

则继续下一阶段的匹配，否则匹配下一对细节点。

- 4) 仿照 3)，计算 α_L ， β_{LM} ， β_{LS} 大小是否分别相同：

$$|P_1.\alpha_L - Q_1.\alpha_L| \leq \varepsilon \quad (5.15)$$

$$|P_1.\beta_{LM} - Q_1.\beta_{LM}| \leq \varepsilon \quad (5.16)$$

$$|P_1.\beta_{LS} - Q_1.\beta_{LS}| \leq \varepsilon \quad (5.17)$$

- 5) 将 F_{LM} 与 F_{LS} 求和进行类似 3) 和 4) 的判断，若满足条件则将匹配三角个数 $InitialScore$ 加 1。

最后统计每一幅样本指纹图像与待识指纹图像的 $InitialScore$ ，达到阈值 T_{max} 时，就认为两枚指纹匹配成功，不需要进行全局匹配；若 $InitialScore < T_{min}$ ，则直接剔除该样本；若 $T_{min} \leq InitialScore \leq T_{max}$ ，则需要再进行全局匹配。

5.6 全局匹配

经过初匹配后，指纹库样本集已经得到了一定的筛选，定义为 set 。并且，

针对 set 中的每一幅图像都存储了候选的参考点。根据文献[54]中经典的全局匹配方法，本文的具体实现如下：

1) 从 set 中选择一幅图像 Q' ，对 P 和 Q' 的参考点对 P_1 和 Q_1 计算旋转因子 r ， trx ， try ，进行配准。见公式(5.18)，(5.19)，(5.20)，令 P 中的每一个细节点 (x, y) 以 Q_1 为中心旋转 r 角，并沿 X 轴方向平移 trx ，沿 Y 轴方向平移 try 后对应得到 P' 。

$$r = Q_1.\theta_o - P_1.\theta_o \quad (5.18)$$

$$trx = Q_1.x_o - P_1.x_o \quad (5.19)$$

$$try = Q_1.y_o - P_1.y_o \quad (5.20)$$

P' 的坐标表达式为：

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(r) - \sin(r) \\ \sin(r) + \cos(r) \end{bmatrix} \begin{bmatrix} x - P_1.x_o \\ y - P_1.y_o \end{bmatrix} + \begin{bmatrix} trx \\ try \end{bmatrix} + \begin{bmatrix} P_1.x_o \\ P_1.y_o \end{bmatrix} \quad (5.21)$$

2) 将配准后的 P' 和 Q' 进行匹配运算。该过程类似于快速匹配，不同的是匹配分数的计算方法。匹配步骤如下：

A、计算两个细节点的方向场夹角：

$$\theta_{dis} = |P_1.\theta_o - Q_1.\theta_o| \quad (5.22)$$

$$\theta_{dis} \leq \varepsilon \quad (5.23)$$

若满足公式(5.23)则继续进行 B。

B、计算两个细节点的坐标差值：

$$x_{dis} = |P_1.x_o - Q_1.x_o| \quad (5.24)$$

$$y_{dis} = |P_1.y_o - Q_1.y_o| \quad (5.25)$$

$$x_{dis} \leq \varepsilon \quad y_{dis} \leq \varepsilon \quad (5.26)$$

若满足公式(5.26)则将匹配对数 $matchnum$ 加 1，并标记样本特征向量中的当前点为已匹配对象，下次不再进行计算。

C、匹配度 $match$ 记录 B 中匹配成功的记录，大小由 θ_{dis} 、 x_{dis} 和 y_{dis} 共同决定：

$$match = (\varepsilon - \theta_{dis}) + \left(\varepsilon - \sqrt{x_{dis}^2 + y_{dis}^2} \right) \quad (5.27)$$

再返回 A 计算下一对参考点的匹配度。

D、根据 C 中累加得到 $match$ 和 B 中累加得到的 $matchnum$ 求取两幅图像的匹配分数：

$$MatchScore = 200 * match * matchnum / (P.pointnum + Q.pointnum)^2 \quad (5.28)$$

考虑到快速匹配中获得的三角形匹配个数 $trianglenum$ 有助于提高匹配准确度，因此，匹配分数的最终公式为：

$$MatchScore = \left(200 * match * \frac{matchnum}{(P.pointnum + Q.pointnum)^2} \right) + \frac{trianglenum * 100}{P.pointnum + Q.pointnum} \quad (5.29)$$

然后返回步骤 1)，计算下一对细节点为配准基点时所得到的 $MatchScore$ ，

直到全部计算完毕后得到最大的匹配分数，若 $MatchScore$ 大于 P 与其他样本图像比对计算的结果，则认为待识别指纹与数据库中的样本匹配，返回指纹样本索引，否则，认为数据库中无匹配指纹。

5.7 实验结果及分析

5.7.1 时间复杂度分析

本文算法经过两级结构相似度计算获得了最终的匹配结果，假设指纹数据库中有 K 个样本，每个样本图像的特征向量是 M 维，输入指纹是 N 维。在初匹配计算相似度时，任意一对参考点的计算量均为常数时间，所以，相应的时间复杂度为 $O(KMN)$ 。经过初匹配的筛选后，如果还需要进行全局匹配，则样本数量缩减到 K' ，参考点对的个数为 $P(P \ll MN)$ ，令其他相同处理花费的时间是 W ，相似度的计算仍然是常数时间，那么全局匹配的时间复杂度就是 $O(K'PW)$ 。对于文献[54]中的方法，直接进行全局匹配的时间复杂度是 $O(KMNW)$ ，由此可见，本文算法在时间上具有较好的优势。

5.7.2 实验结果分析

为了验证本文算法的性能，将指纹采集仪采集的 960 幅分别属于 60 枚手指的指纹图像作为实验对象。其中 60 幅是质量较好，面积完整的样本图像，其他 900 幅分别存在不同力度按压，不同角度旋转，干手指和湿手指采集等情况，更贴近实际采集结果。

对指纹识别算法测试的方案分为两部分，一是遵循第二章所介绍的 FVC 测试方法：首先对库中每个手指的 16 枚指纹进行内部多级匹配，总共需要 $60 * C_{16}^2 = 7200$ 次，拒识率 $FRR = \text{拒绝识别次数} / 7200$ 。然后从库中为每个手指随机抽取一幅采集面积足够的图像，构成一个总共 60 个样本的图像子集，从中任选一幅指纹图像与其他 59 幅进行交叉匹配，总共需要 $C_{60}^2 = 1770$ 次，误识率 $FAR = \text{错误识别次数} / 1770$ 。

分别就 FAR 、 FRR 与运行时间对本文算法与文献[54]中的方法进行对比，如表 5-1 所示。

表5-1 本文算法与文献[54]对比结果

| 算法 | 误识率 (%) | 拒识率 (%) | 匹配时间 |
|--------|---------|---------|----------|
| 文献[54] | 0.11% | 3.28% | 450~2000 |
| 本文算法 | 0.06% | 1.59% | 90~600 |

第二部分是基于实际的自动识别系统中都是 1:N 异源指纹匹配模式的方式，所以将 900 幅指纹图像作为待识图像，分别与 60 幅样本图像进行比对，实验结果显示 316 幅可以直接通过初匹配得到正确的结果，584 幅需要继续进行全局匹配才能得到正确的匹配。另有 11 幅图像无法识别，7 幅图像因为其真正的同源指纹在初匹配时被剔除而被匹配到了错误的样本，最终的识别正确率如表 5-2。

表5-2 1:N 异源匹配结果

| 算法 | 误识率 (%) | 拒识率 (%) | 正确率 (%) |
|------|---------|---------|---------|
| 本文算法 | 0.78% | 1.12% | 98.1% |

上表中的匹配时间中包含了读取数据库的时间，所以实际匹配的时间会更小。由表中的统计数据分析可知，本文提出的两级匹配的整体性能是高于文献[54]的，改进后的算法使匹配精度和时间复杂度达到了可接受的程度。但是由于实验条件有限，本次实验只进行了近五万次的匹配，在实际应用中，情况是复杂多变的，所以还有待用大规模指纹样本库进行性能测试和压力测试。

5.8 本章小结

本章深入研究了指纹图像匹配的相关算法，在前人工作的基础上提出了一种两级匹配的方法。首先对每一个细节点定义具有良好区分度的子结构，包括外接全等三角形和近邻点组成的三角形，具体地反映了细节点周围的方向场和相对位置关系。然后根据子结构依次比对边角关系，两者的结合有效地降低了局部邻域信息错误匹配的概率。最后，根据筛选的结果，决定是否再进行全局匹配。由于初匹配帮助剔除了相当多的样本和不符合条件的参考点，所以加快了全局匹配的处理速度。实验结果表明，本文的算法在运行速度和准确性方面跟文献相比都有了一定的提高。

此页不缺内容

第六章 自动指纹识别系统软件的设计

6.1 软件平台搭建

为了更直观地对自动指纹识别系统中的关键算法进行评估,本文在研究了指纹预处理、旋转校正、分类和匹配多种算法的基础上,开发了基于 VS2010 开发环境和 MYSQL 数据库的 AFIS 软件。MFC 应用程序开发环境封装了大部分 API 函数并提供一个应用程序框架。利用应用程序向导建立对话框程序框架,资源编辑器搭建用户界面,类向导创建成员变量、消息机制和处理函数,大大加快了软件开发过程,缩短开发的周期^[62]。MYSQL 是一种多线程、多用户的、跨平台的轻量级的小型数据库管理系统,具有源码开源、速度快、体积小的优点^[63]。软件平台的开发为自动指纹识别系统的进一步研究奠定了基础。

- 测试平台: Windows 7
- 处理器: Intel®Celeron®, 2.40GHz
- 内存: 2.00GB

6.2 自动指纹识别系统软件整体设计

6.2.1 基于界面结构划分

根据需求分析,自动指纹识别系统软件平台主要实现算法研究和算法应用功能。图 6-1 是最终软件平台界面示意图,A 区域是算法研究模块,用于单步调试验证各部分算法效果;B 区域是算法应用模块,用于四个主要功能的单独操作;C 区域是图像读取和保存模块;D 区域负责显示图像效果及图像处理结果。各功能模块的总体设计如图 6-2。



图6-1 自动指纹识别系统软件平台界面

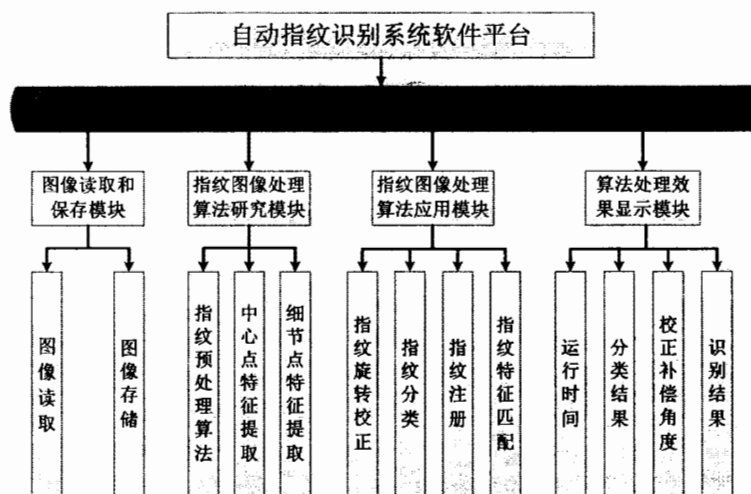


图6-2 自动指纹识别系统软件平台的功能划分

1. 图像读取和保存模块：主要用于打开采集到的原始指纹图像，并显示在区域 D1 中；保存功能是记录区域 D2 中显示的中间处理结果图像。
2. 指纹图像处理算法研究模块：主要整合了预处理算法和特征提取的多个关键算法，总共十四个步骤，可以分步操作，帮助查看处理的中间结果。
3. 指纹图像处理算法应用模块：包括四个功能，分别整合了相关的算法，包括指纹旋转校正、指纹分类、指纹匹配过程中的注册和匹配。
4. 算法处理效果显示模块：给出各种算法的处理效果及运行时间，主要包括旋转校正需要补偿的角度、分类结果及匹配是否成功等，定性定量地评估算法性能及效果。

6.2.2 基于面向对象的类划分

根据面向对象的思想，本文的软件平台涉及到的算法、图像数据都可看作是对象，并在软件实现中抽象为 C++ 的类，以此来表示各个对象的属性和方法。综上，基于单一职责原则，即就一个类而言，应该仅有一个引起它变化的原因^[64]，本文主要划分为三类，如表 6-1。

表6-1 指纹识别系统中使用的类

| 类名 | 主要功能 |
|-------|---|
| 图像表示类 | 主要是由 OpenCV 库封装，定义了图像二维矩阵的数据存储方式和一般的处理函数。 |
| 算法类 | 主要实现预处理算法、特征提取、旋转校正、分类和匹配算法。 |
| 对话框类 | 软件框架类，主要显示各算法的处理结果。 |

MFC 本身是基于消息机制的，本文在Dlg类中定义各个算法对应的消息相应函数^[53]，具体执行流程如图 6-3。

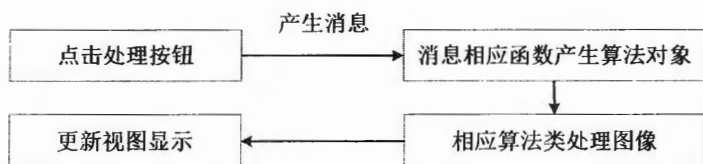


图6-3 从界面到算法类的消息处理流程

6.3 系统平台各功能模块详述

6.3.1 图像读取保存模块

由于MFC并未提供封装好的图像类，并且本文是基于OpenCV库实现部分图像算法，所以利用IplImage存储图像数据。但是，MFC的Picture控件不能直接加载IplImage格式的图像，为此，本文实现了HBITMAP IplImage2hBitmap(IplImage* pImg)函数，即后台操作的图像格式是IplImage，当需要显示到界面控件上时，就转换成HBITMAP格式。

每当打开一幅指纹图像时，便产生一个IplImage图像指针，并删除上一幅图像的指针；当需要将处理的中间结果图保存时，即可选择保存路径并成功将区域D2中显示的图像保存为BMP格式位图。

6.3.2 指纹图像处理算法研究模块

指纹图像处理算法研究模块主要包括预处理算法和特征提取。其中，预处理算法区域总共包含十个按钮选项，分别对应：直方图均衡化、图像规格化、高斯滤波、指纹图像分割、方向图求取、脊线增强、二值化、二值化去噪、细化脊线和细化去噪。为了防止用户在没有打开图片时误操作，所以在未打开一幅新的指纹图像之前十个按钮都处于灰化不可点击状态。同时，预处理算法是按这十个步骤依次对图像进行处理，当第一步图像直方图均衡化操作完成后，“归一化”按钮方能解除灰化。

特征提取算法则由全局特征提取和局部特征提取两部分组成，又可细分为奇异点提取、去伪奇异点、细节点提取和去伪细节点四个部分。与预处理相似，四个相应的按钮初始时都为灰化状态。

以图 6-4 为例，说明算法顺序是当前一个步骤的输出有效，后一个步骤的输入满足要求时，其对应按钮为启动状态。

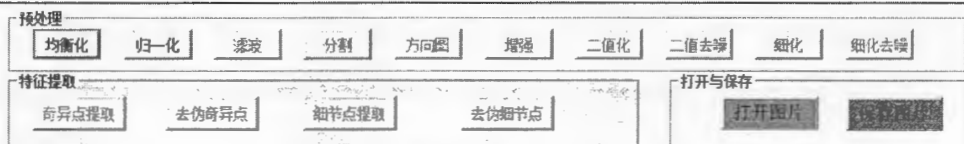


图6-4 算法处理顺序示意图

6.3.3 指纹图像处理算法应用模块

该模块的四个功能相互独立，都集成了各自需要的预处理和特征提取操作。当选择任一功能时，其他所有选项都处理灰化状态。旋转校正主要用于倾斜图像的校正，输出图像的同时提示用户倾斜的角度，以便于验证和调试；分类功能是

对指纹进行多级分类，并显示最终的分类结果；注册和特征匹配主要涉及到指纹特征向量的存储和提取。其中分类、注册和特征匹配都需要写入和读取数据库，其结构关系如图 6-5。

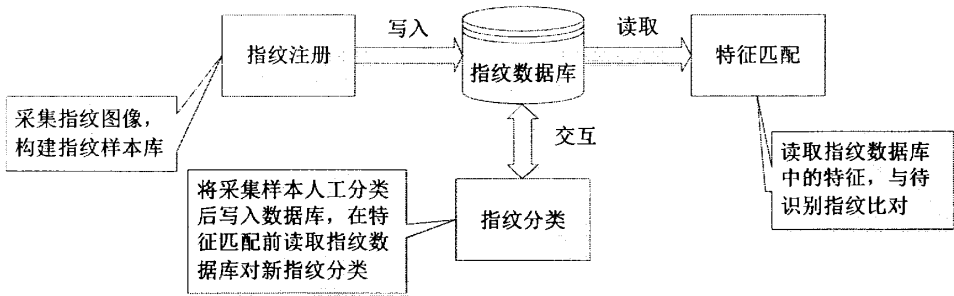


图6-5 算法应用模块数据流图

考虑到大规模指纹库的应用背景，本文采用 MYSQL 管理数据库，数据库名称为 AFIS_Feature，保存样本指纹的特征向量集。

数据表是数据库逻辑结构的核心部分，只有通过数据库对象建立与应用模块对象的映射关系之后，才能进行应用系统的设计，只有好的数据表设计，才能完全满足系统的各项要求，从而构建强健稳定的系统^[65]。下面对一些重要的数据表进行介绍。

1. 用户指纹注册表（表 6-2）

主要对指纹的 ID，所属用户的姓名和 ID 进行记录。

表6-2 指纹注册用户信息表

| 列名 | 数据类型 | 数据长度 | 允许空值 | 描述 |
|------------|---------|------|------|----------|
| finger_ID | INT | 4 | 否 | 主键，自增长 |
| owner_name | VARCHAR | 20 | 否 | 指纹注册人姓名 |
| owner_ID | INT | 4 | 否 | 指纹注册人 ID |

2. 指纹分类信息表（表 6-3）

主要记录指纹在三级分类后所属的类别。

表6-3 指纹图像类别信息表

| 列名 | 数据类型 | 数据长度 | 允许空值 | 描述 |
|-----------------|------|------|------|----------|
| finger_ID | INT | 4 | 否 | 主键，自增长 |
| second_category | INT | 4 | 否 | 指纹二级分类类别 |
| third_category | INT | 4 | 否 | 指纹三级分类类别 |

3. 指纹细节点索引表（表 6-4）

当存储空间十分充足时，可以将完整的特征向量进行存储，表 6-4 是指纹图像的特征向量信息表。

表6-4 完整的特征向量信息表

| 列名 | 数据类型 | 数据长度 | 允许空值 | 描述 |
|---------------|-------|------|------|-----------------|
| finger_ID | INT | 4 | 否 | 主键, 自增长 |
| Type | INT | 4 | 否 | 细节点类型 |
| point_x | INT | 4 | 否 | 分叉点 x 坐标值 |
| point_y | INT | 4 | 否 | 分叉点 y 坐标值 |
| direction | FLOAT | 4 | 否 | 分叉点方向 |
| Trangle1 | FLOAT | 4 | 是 | 全等三角形方向 |
| Trangle2 | FLOAT | 4 | 是 | 全等三角形方向 |
| Trangle3 | FLOAT | 4 | 是 | 全等三角形方向 |
| Neighbor_len1 | INT | 4 | 是 | 近邻三角形长边 |
| Neighbor_len2 | INT | 4 | 是 | 近邻三角形中长边 |
| Neighbor_len3 | INT | 4 | 是 | 近邻三角形短边 |
| Neighbor_ang1 | FLOAT | 4 | 是 | 近邻三角形长边对角 |
| Neighbor_ang2 | FLOAT | 4 | 是 | 近邻三角形长边两点和中心点夹角 |
| Neighbor_ang3 | FLOAT | 4 | 是 | 近邻三角形长边两点和中心点夹角 |
| Neighbor_bri1 | INT | 4 | 是 | 近邻三角形三边穿越脊线数 |
| Neighbor_bri2 | INT | 4 | 是 | 近邻三角形三边穿越脊线数 |
| Neighbor_bri3 | INT | 4 | 是 | 近邻三角形三边穿越脊线数 |

4. 指纹细节点索引表 (表 6-5)

当存储空间相对紧张时, 可以只存储必要的特征信息, 而采取以时间换空间的做法, 实时计算出所需要的匹配特征。表 6-5 是简易的指纹图像特征向量信息表。

表6-5 简易特征向量信息表

| 列名 | 数据类型 | 数据长度 | 允许空值 | 描述 |
|-----------|-------|------|------|-----------|
| finger_ID | INT | 4 | 否 | 主键, 自增长 |
| Type | INT | 4 | 否 | 细节点类型 |
| point_x | INT | 4 | 否 | 分叉点 x 坐标值 |
| point_y | INT | 4 | 否 | 分叉点 y 坐标值 |
| direction | FLOAT | 4 | 否 | 分叉点方向 |

6.3.4 运行结果显示模块

运行结果显示主要为了展示算法处理的效果图和适当的对话框提示, 以便开发者能够更好地评判算法性能。根据不同的功能模块, 选取部分处理结果进行展示, 如图 6-6、图 6-7、图 6-8。



图6-6 算法研究模块细节点提取示意图



图6-7 算法应用模块指纹分类示意图

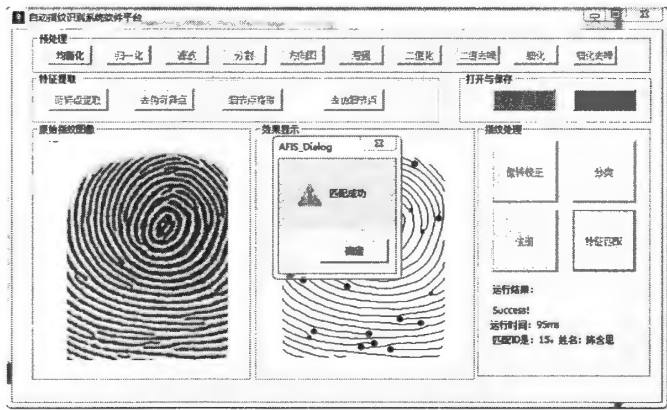


图6-8 算法应用模块指纹匹配示意图

6.4 本章小结

为了测试验证指纹相关图像处理算法是否满足预期的效果，也为进一步提高算法的性能，本章构建了一个基于 Windows 的自动指纹识别系统的软件平台。首先介绍了该平台的总体设计，主要从基于界面划分模块和基于面向对象划分模

块两方面展开说明，随后具体描述了四个模块的具体功能和实现，以及相对应的数据库存储设计，最后展示了算法的处理效果。总得来说，该软件平台实现了友好的人机界面，易于操作，算法扩展性良好，能够有效地展示测试算法性能及运行结果，达到了预定的要求。

此页不缺内容

第七章 总结与展望

自动指纹识别技术一直是国内外学术界的研究热点,经过几十年的不断努力,已取得了显著的成果。随着指纹识别应用深入到民用系统,并不断朝着网络化、智能化、集成化发展,对指纹相关技术算法的准确性和效率提出了更高的要求。目前的指纹识别系统仍然存在着一些问题有待深入探究,如残缺指纹的识别、低质量指纹图像的预处理、奇异点的精确提取及大规模指纹样本的快速分类匹配等。

本文针对目前指纹识别技术所存在的问题,结合数字图像处理、模式识别、人工智能等多方面的知识,深入研究了自动指纹识别系统的相关算法。下面对本人完成的工作进行总结,并对未来的研究内容作出展望。

7.1 工作总结

1. 研究并改进了指纹图像预处理算法中的关键步骤。

在指纹图像预处理环节,对采集到的指纹图像进行了一系列的预处理工作,并着重对图像均衡化、方向图计算和图像增强进行了创新和改进。

首先,针对传统的直方图均衡化,由原来计算整个灰度范围内所有像素点的灰度概率改为只在一定范围内统计,在最大程度降低脊线灰度的同时扩大谷线的灰度范围,如此能够更好地提高脊线与谷线的对比度,增加前景区域灰度方差。其次,在计算指纹方向场时,为了兼顾到时间效率和奇异点周围方向多变的特性,设计了一种结合块方向图和点方向图各自优点的算法。该方法在快速获得方向场的基础上,保留了奇异区附近的方向细节。最后,利用简化的十字交叉方向滤波器代替文献中的 Gabor 滤波法对指纹图像进行增强。实验结果表明,经过预处理算法操作后,指纹图像纹路清晰,特征明显,为后续算法奠定了良好的基础。

2. 研究并改进了奇异区中心点的提取方法。

在中心点的提取上,对传统的基于 Poincare 的索引算法进行改进。首先,根据方向场结果对中心点进行粗提取,即定位包含奇异点的区域,然后对子块进行两次 Poincare 索引,获得初步的提取结果,最后采用完善的去伪奇异点方法除去错误的中心点。其中,去伪方法主要结合了指纹本身的特性,排除以下几种情况:中心点附近不符合归域化规律、多个中心点距离较近、背景区噪声干扰、附近纹线曲率较小、附近方向场不符合拱形分布规律等。最终的实验结果表明,此方法能够准确地提取中心点信息,具有很强的抗噪能力。

3. 研究并提出了一种基于旋转校正图像和中心点特征的的三级指纹分类方法。

本文的指纹分类方法主要包括旋转校正、中心点提取和分类三个步骤。首先,由于指纹采集仪采集面积有限,不能直接采用经典的奇异点分类算法,所以本文在提取中心点之前对有旋转的指纹进行旋转校正以获得更多更可靠的分类特征。主要是采用拟合中位线法和最小外接椭圆法获得需要补偿的角度,再通过仿射变换进行校正。然后,提取奇异区的中心点及属性信息。最终,利用中心点的数目、方向和分布规律完成前两级分类,再根据螺径、纹线附近拱形方向分布等特征进行第三级分类。其中,一级分类和二级分类都属于排斥性分类方法,类间相关性低,三级分类属于连续分类,分类数目较多。

4. 研究并提出了一种两级指纹匹配方法。

针对传统点模式匹配方法重复性计算量过大的不足,本文提出了一种结合了初匹配和全局匹配的两级指纹匹配方法。首先,通过对指纹中的细节点定义具有良好区分度的子结构来帮助构建特征向量。然后,比对此子结构的边角关系判断是否匹配,如若初匹配失败,则进入全局匹配。最后,根据筛选的结果完成全局匹配的配准和比对工作。实验结果表明,由于初匹配剔除了部分不匹配的样本和参考点对,所以在不影响识别正确率的前提下大大加快了处理速度。

5. 研究并开发了简易的自动指纹识别系统软件。

为了更好地研究指纹识别系统的关键算法,本文基于 VS2010 和 MYSQL 编写构建了自动指纹识别系统软件平台,该平台包括了指纹图像打开和保存模块、指纹图像算法研究模块、指纹图像算法应用模块和指纹图像处理结果展示模块。从面向对象角度将各个算法、界面类、图像数据视为对象,遵循单一职责原则,易于开发。实践证明该软件维护方便,扩展性好,可用于对自动指纹识别算法的进一步研究。

7.2 展望

自动指纹识别技术是一个颇具挑战的研究领域,虽然本文对几个关键算法进行了一定的研究,但由于时间和工作量的关系,还存在一些需要进一步深入研究和完善的方面:

1. 研究低质量小面积的指纹图像相关算法。指纹采集在没有人为干预的情况下,经常会获得不理想的图像,比如:很多指纹存在断裂、划痕等情况,虽然可以通过图像增强将脊线进行连接,但仍会带来一定的伪端点,另外,目前非线性形变指纹的配准方法并不十分完善,来自不同手指的指纹可能会被误认为相似,干扰识别判断,同时还存在采集区域受污染或者手指较潮湿等因素带来的指纹图像模糊不清的情况,即使提取到特征点,其可信度也非常低,这增大了匹配的难度。因此,需要进一步探索适合不同情况的指纹图像质量评价指标,以在采集时就对图像质量进行有效评估,必要时要求重新采集合适的指纹图像。

2. 文中的指纹旋转校正算法虽然易于实现,但仍然存在不足:仿射变换需要进行插值运算,这是一种用时间换取准确率的做法,所以下一步可以只旋转方向场及用于后续算法的数据,而不必对整幅灰度图操作。另外,在时间复杂度允许的情况下,可以加入反馈处理机制,建立方向场校验模型,针对校正不当的图像重新计算旋转角度。

3. 针对大容量指纹样本数据库的分类检索研究还需要进一步深入探索,通过大量实验找到可靠的分类特征,将类别进行更细致的划分,也可以将信息检索和数据挖掘理论引入到海量指纹数据的应用中。

4. 目前开发的软件平台是在 PC 机上基于小样本数据库运行的,虽然分类和识别效果良好,但是仅限于实验室研究,并未推广至实际的产品应用。接下来在扩大指纹样本的规模,继续改进算法的效果同时,可以考虑移植到嵌入式系统中。由于嵌入式系统可用资源相对于计算机比较有限,因此需要重新考虑算法对资源的占用情况,即在保证识别速度的前提下,仍然达到较高的正确率。

参考文献

- [1] Clarke R. Human identification in information systems: Management challenges and public policy issues [J]. Information Technology and People, 1994, 7(4): 6-35.
- [2] J.L.Wayman. Fundamentals of biometric authentication technologies [J]. International Journal of Image and Graphics, 2001, 1(1): 93-113.
- [3] Biometrics Market and Industry Report 2007–2012. GLOBAL IDENTITY MANAGEMENT [EB/OL]. <http://findbiometrics.com/>.
- [4] Osterburg J., Parthasarathy T., Raghaven T., Sclove S. Development of A Mathematical Formula or the Calculation of Fingerprint Probabilities Based on Individual Characteristic [J]. Journal American Statistic Association, 1977, 72: 772-776.
- [5] 池景臣. 用于考勤系统的指纹识别方法研究[D]: [硕士学位论文]. 黑龙江: 哈尔滨工程大学, 2008.
- [6] Zhou J., Chen F., Wu N., Wu C. Create Detection from Fingerprint Images and Its Application in Elderly People [J]. Pattern Recognition, 2009, 42(5): 896-906.
- [7] H.C. Lee and R.E. Gaensslen. Advances in fingerprint technology. 2nd edition [M]. New York: Elsevier, 2001.
- [8] Ashbaugh D. R. Quantitative-qualitative Friction Ridge Analysis: An Introduction to Basic and Advanced Ridgeology [M]. Boca Raton: CRC Press, 1999.
- [9] Cummins H., Midlo C. Finger Prints, Palms and Soles: An Introduction to Dermatoglyphics [M]. New York: Dover, 1943.
- [10] A. Moenssens. Fingerprint Techniques [M]. London: Chilton, 1971.
- [11] Faulds Henry. On the Skin-furrows of the Hand [J]. Nature, 1880, 22: 605-605.
- [12] 王玮. 自动指纹识别系统关键技术研究[D]: [博士学位论文]. 重庆: 重庆大学, 2007.
- [13] 田捷, 陈新建, 张阳阳等. 指纹识别技术的新进展[J]. 自然科学进展, 2006, 16(4): 400-406.
- [14] 国务院. 国家中长期科学和技术发展规划纲要(2006-2020年) [Z]. 2006.
- [15] 陈晖. 结合方向场特征的扭曲指纹图像识别技术研究[D]: [博士学位论文]. 湖南: 国防科学技术大学, 2011.
- [16] 机器识别指纹与人工识别指纹的区别与联系. 中国数字科技馆[EB/OL]. <http://www.cdstm.cn/>.
- [17] 张明志. 基于微特征的指纹识别算法研究[D]: [硕士学位论文]. 福建: 厦门大学, 2009.
- [18] 柴晓光, 岑宝炽. 民用指纹识别技术[M]. 北京: 人民邮电出版社, 2004.
- [19] 薛俊韬. 基于大规模数据库的多级指纹分类研究[D]: [博士学位论文]. 天津: 天津大学, 2010.
- [20] 黄玮, 林明琚, 刘持平. 汉族人指纹正常值的统计与分析[J]. 江苏警官学院学报, 2006, 3(21): 154-159.
- [21] 付莉娜. 指纹识别算法的研究与优化[D]: [硕士学位论文]. 陕西: 西安科技大学, 2012.

- [22]特殊聚合物相助识别汗腺孔排布. 科学网[EB/OL]. <http://paper.sciencenet.cn/>
- [23]肖蓉. 自动指纹识别技术的研究[D]: [硕士学位论文]. 江苏: 南京理工大学, 2007.
- [24]何昕, 李晓华, 周激流. 一种自适应多阈值直方图均衡方法[J]. 计算机工程, 2011, 37(17): 206-210.
- [25]余锦华, 陈建华, 施心陵. 指纹图像的预处理[J]. 计算机工程与设计, 2004, 25(12): 2210-2212.
- [26]周俊, 罗挺, 路翔, 王冰. 一种基于拉伸变换函数的指纹图像增强算法[J]. 计算机系统应用, 2009, 3: 141-143.
- [27]Yu Shiaw Shian, Tsai Wen Hsiang. A new thinning algorithm for gray-level images by the relaxation technique [J]. Pattern Recognition, 1990, 23(10): 1067-1076.
- [28]Rafael C. Gonzalez, Richard E. Woods. 数字图像处理(第二版)[M]. 北京: 电子工业出版社, 2007.
- [29]李昊, 傅曦. 精通 Visual C++指纹模式识别系统算法及实现[M]. 北京: 人民邮电出版社, 2008.
- [30]Hong L., Wang Y F, Jain A K. Fingerprint Image Enhancement: Algorithm and Performance Evaluation [J]. IEEE Transaction on Pattern Analysis and Machine Interlligence, 1998, 2(8): 777.
- [31]向锐. 面向低质量指纹的图像增强算法研究[D]: [硕士学位论文]. 重庆: 西南政法大学, 2008.
- [32]韩涛. 指纹图像预处理与特征提取算法研究[D]: [硕士学位论文]. 湖北: 华中科技大学, 2005.
- [33]陈若珠, 田守军, 李战明, 吕强. 一种健壮的指纹方向图校正算法[J]. 电子测量技术, 2009, 32(2): 143-146.
- [34]孟建民. 嵌入式指纹采集系统研究[D]: [硕士学位论文]. 江苏: 南京航空航天大学, 2008.
- [35]Lawrence O'Gorman. Matched Filter Design for Fingerprint Image Enhancement [J]. Proceeding of International Conference on Acoustics, Speech, and Signal Processing. 1988: 916-919.
- [36]张新明, 党留群, 徐久成. 基于十字滑动窗口的快速自适应图像中值滤波[J]. 计算机工程与应用, 2007, 43(27): 37- 43.
- [37]高婧婧. 指纹识别预处理算法研究[D]: [硕士学位论文]. 四川: 电子科技大学, 2004.
- [38]王玮, 李见为, 张腾. 指纹图像的预处理算法[J]. 计算机应用, 2004, 24(5): 72-75.
- [39]邓廷权, 唐东平. 数学形态学视域下指纹图像细化算法高效研究[J]. 湘潭大学自然科学学报, 2012, 34(3): 93-97.
- [40]颜沙. 大规模指纹库的快速指纹识别算法研究[D]: [硕士学位论文]. 湖北: 华中科技大学, 2013.
- [41]刘通. 自动指纹分类算法研究[D]: [硕士学位论文]. 北京: 清华大学, 2001.
- [42]pitch yaw roll 的区别. 网易博客[EB/OL]. <http://blog.163.com/>.
- [43]Cao K., Yang X., ao X., Li TP., Zhang Y., Tian J. Combining Features for Distorted Fingerprint Matching [J]. Journal of Network and Computer Applications, 2010, 33(3): 258-267.

- [44]K. Torii, N. Okamoto. Extraction Algorithm of Principal Lines and Rotation Method for Non-Contacting Fingerprint Authentication [J]. CCECE-2004, 2004: 988-991.
- [45]Gary Bradski, Adrian Kaebler. 学习 OpenCV [M]. 北京: 清华大学出版社, 2009.
- [46]李双梅. 指纹自动分类技术的研究与实现[D]: [硕士学位论文]. 江苏: 南京理工大学, 2007.
- [47]牛重, 指纹旋转情况判定[J]. 信息技术, 2012, 8: 164-171.
- [48]张祖洸, 杨永明等. 指纹奇异点精确定位新方法[J]. 计算机工程与应用, 2012, 48(32): 203-206.
- [49]翁大伟, 尹义龙等. 基于 Gaussian-Hermite 矩和改进的 Poincare Index 的指纹奇异点提取[J]. 计算机研究与发展, 2008, 45(11): 1975-1977.
- [50]梅园, 孙怀江等. 一种有效的指纹奇异点提取方法[J]. 计算机工程与应用, 2008, 48(28): 1-3.
- [51]A. Lumini, D. Mario, D. Maltoni. Continuous Vs Exclusive Classification for Fingerprint Retrieval[J]. Pattern Recognition Letters, 1995, 3: 578-585.
- [52]杨小冬, 宁新宝等. 基于纹线跟踪的指纹分类方法[J]. 计算机工程, 2005, 31(7): 170-173.
- [53]彭曦. 基于大容量指纹库的自动指纹识别系统研究[D]: [硕士学位论文]. 湖南: 中南大学, 2013.
- [54]Jain A K., Hong L., Bolle R. On-line Fingerprint verification[J]. IEEE Transaction on Pattern Analysis and Machine Intelligence, 1997, 19(4): 302-314.
- [55]A. Ross, J. Reisman, A. Jain. Fingerprint Matching Using Feature Space Correlation[J]. Biometric Authentication Lecture Notes in Computer Science, 2006, 2359: 48-57.
- [56]X.D. Jiang, W. Y. Yau. Fingerprint Minutiae Matching Based on the Local and Global Structures [C]. In: Proceedings of the 15th International Conference on Pattern Recognition, 2000: 1037-1040.
- [57]杨栋, 杨公平, 尹义龙, 张利明. 基于多级结构相似度的快速指纹匹配算法[J]. 山东大学学报, 2010, 40(2): 11-18.
- [58]Janm D., Nal N., et al. Binary Tree Based on Linear Time Fingerprint Matching[C]. Proceedings of International Conference on Image Processing. IEEE Computer Society Press, 2006: 309-312.
- [59]尹义龙, 张宏伟, 刘宁. 基于三角化的指纹匹配方法[J]. 计算机研究及发展, 2005, 42(9): 1622-1627.
- [60]K. Nandakumar. Fingerprint Matching Based on Minutiae Phase Spectrum[C]. ICB, 2012 5th IAPR International Conference on IEEE, 2012: 216-221.
- [61]吕玉华. 基于结构特征匹配的指纹识别算法研究[D]: [硕士学位论文]. 湖南: 湖南大学, 2008.
- [62]王占全, 徐慧. 精通 Visual C++数字图像处理技术与工程案例[M]. 北京: 人民邮电出版社, 2009.
- [63]Ben Forta. MySQL 必知必会[M]. 北京: 人民邮电出版社, 2009.
- [64]侯杰. 大话设计模式[M]. 北京: 清华大学出版社, 2007.
- [65]周辉. 基于 WebGIS 与车载移动视频的智能车辆监控系统[D]: [硕士学位论文]

文]. 江苏: 东南大学, 2011.

致谢

本毕业设计是在导师、同学、朋友、家人等的帮助下完成的。回顾硕士两年多来的学习生活,我所取得的成绩和进步,离不开良师益友及亲人的鼓励和帮助,在此,我真心地向他们表示最诚挚的感谢。

本文是在叶桦教授的精心指导下完成的。从论文的选题到算法各个模块的实现及改进,到最终论文的定稿,无处不浸透着叶老师的心血。论文完成的过程中,我遇到了很多的困难,经历了多次瓶颈阶段,例如指纹分类算法的提出及优化,特征匹配如何提高速度和精度等,在这些问题上我困惑了很长时间,是叶老师不断地鼓励我,支持我,同我一起分析问题的根源,寻求合理的解决方案。一路走来,叶老师像一座灯塔,给以我方向性的指引,让我不再被动地面对困难,而是勇敢地想办法去克服它们,最终顺利完成了我的项目和论文。叶老师渊博的学识、温和谦逊的态度、严谨踏实的工作作风、诲人不倦的治学精神潜移默化地影响着我,让我受益匪浅。特别是他积极乐观的人生态度和宽厚待人的处事方式,让我更加乐观地去面对生活,与人相处也更为融洽。在此,谨向叶老师致以我最崇高的敬意和最衷心的感谢。

本课题的完成,也离不开实验室仰燕兰师姐及许多同学的热情帮助。仰燕兰师姐有着渊博的学术知识和丰富的科研经验,每当遇到难题时,她都会挺身而出,提出非常有见地的想法,耐心帮我分析,借此向仰师姐表示我真诚的谢意。

此外,我还要感谢我的其他同门吴晓佳、高菊、张芸、张国颖、陈华成、方长青、吴静,以及实验室的所有师弟师妹,是你们陪伴我渡过了难忘的硕士研究生生涯,能在这样一个温馨的集体中成长,我感到无比幸福。

最后,我要感谢我的家人和朋友,感谢你们给予我的无微不至的关怀与鼓励,让我积极进取,不断前进。在今后的生活中,我一定不会辜负你们的期望,不断奋斗,勇往直前。

最后,衷心感谢在百忙之中评阅论文和参与答辩的各位老师。在毕业后,我一定会运用所学的知识努力工作,回报社会,回报家庭,回报关心和支持我的老师和领导,为国家的繁荣和社会的发展做出应有的贡献。

此页不缺内容

作者在攻读硕士学位期间发表的论文和获奖情况

论文:

第一作者. 微操作机器人系统上位机软件的设计. 信息技术与信息化, 2013(5): 78-81.

第一作者. 指纹图像的旋转校正与分类. 计算机科学与应用, 2014, 4(5): 85-94.

获奖:

硕士研究生国家奖学金. 中华人民共和国教育部颁发, 2013.

研究生数学建模竞赛二等奖. 中华人民共和国教育部学位与研究生教育发展中心颁发, 2013.

东南大学研究生三好学生. 东南大学颁发, 2013.

东南大学社会工作单项奖. 东南大学颁发, 2013.

东南大学自动化学院优秀党员奖. 东南大学自动化学院颁发, 2014.