

摘 要

生物识别技术被认为是自动身份识别的最终技术。其中自动指纹识别技术是生物识别中最成熟的生物识别技术。指纹的唯一性和不变性被用来作为鉴别个人的主要基础并有着美好的未来。自动指纹识别系统是基于计算机来进行指纹识别的技术，可以方便、高效、可靠地应用在金融安全、数据加密、电子商务等各个领域，并将在我们的生产和生活中发挥越来越重要的作用。

由于指纹图像的噪声等各种因素的影响，指纹识别一直存在识别率不高、运算速度较慢、指纹数据库难以建立等各种问题。针对上述问题，作者从指纹图像预处理算法、特征提取、匹配、数据库的建立等环节入手，建立一个现实可用的自动指纹识别软件。

准确可靠地提取细节特征是自动指纹识别实现的前提和基础，而细节特征提取的准确性严重依赖于指纹图像的质量，指纹图像的预处理主要是指指纹图像增强算法的研究。图强增强算法首先通过把图像进行分割成大小相同的小块，以该小块中心点的脊频和方向来代替该小块每一点的脊频和方向，这样使用量化后Gabor滤波器的脊频和方向的方式改进传统的指纹图像增强算法，最后具体在软件中进行了实现。特征提取主要是提取指纹的细节特征及其位置，再对每个细节特征进行验证，对存在的伪特征点进行滤除。采用优点突出的点模式匹配算法来实现指纹的匹配。实现指纹特征的存储功能是基于xml建立的数据库并描述这种方法的优点及其实现方式。

最后对提出的预处理、特征提取等算法进行软件实现，并参考国际公认的评价体系对软件进行评估，测试结果表明以上指纹识别算法可有效的增强指纹图像，能准确地提取指纹特征点，准确的进行特征点匹配。

关键词:指纹识别 Gabor滤波 二值化 细化 匹配

ABSTRACT

The biometric technology is considered as the end technology of automatic identity verification technology. And automatic fingerprint identification technology is the most mature one of the biometric technology, the uniqueness and invariableness of fingerprints is used as the main basis to identify a man and it has a great future. Automatic fingerprints identification system which is a technology based on computer could be deployed to financial security, data decryption, e-commerce and so on conveniently and efficiently, and it will gain more important place in our daily life.

Because of noises and other influencing factors, the automatic fingerprint identification technology has many problems like low identification rate, low speed, hard to build database and so on. The author bring forward a software which has many improvements on fingerprints image preprocess, minutia extraction and database to solve the upper problems. The main improvements on fingerprints image preprocess are to eliminate the noises to clarify the image, get back the ridge line, then thin the ridge line to one pixel width line. These next steps are image normalization, direction calculation, ridge frequency calculation, image enhancement, binarization and postprocess. Minutiae extraction is to extract the minutiae details like position, then eliminate the fault point. This paper give a glance of algorithms already used today and tell the advantages and disadvantages of those algorithms, Finally this paper use matching based on minutiae to judge whether two fingerprint come from the same finger or not. This paper use xml to build the database and tell the strength of doing this.

Finally this paper use a evaluation system which is already accepted by the world to evaluate the software, and the test results shows that this software could enhance the fingerprint efficiently, extract the minutiae point correctly and match the minutiae point correctly.

Keywords: Fingerprint verification Gabor filter Binary Thinning Matching

独创性声明

秉承学校严谨的学风与优良的科学道德，本人声明所呈交的论文是我个人在导师的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，不包含本人或其他用途使用过的成果。

本学位论文成果是本人在广东工业大学读书期间在导师的指导下取得的，论文成果归广东工业大学所有。

申请学位论文与资料若有不实之处，本人承担一切相关责任，特此声明。

指导老师签字：

曾建谋

论文作者签字：

杜宝雷

2007年06月15日

第一章 绪论

1.1 国内外指纹识别系统的研究历史与应用现状

1.1.1 国外指纹识别研究的历史与现状

安全性是许多系统要考虑的问题。常用的身份认证的方式有三种:基于口令的认证方式,基于介质的认证方式和基于生物特征的认证方式。口令认证方式是最常用的一种技术,口令的易泄露性和易丢失性使它存在严重的安全问题。采用磁卡等的基于介质的认证方式,具有硬件加密功能,有较高的安全性,但仍存在介质遗失、被盗和被仿造的可能。上述两种认证方式都是传统的身份识别技术,是通过存在于身份外的某物来辨认和鉴别人的身份的,根本无法对需要认证的人本身进行直接辨认,很难实现真正安全、可靠的身份识别。而基于生物特征的认证方式,是采用每个人的身体的某个特征具有与其他人不同的唯一性和在一定时期内不变的稳定性,且具有不可复制的特性进行身份识别,因此生物识别技术成为目前最为方便与安全的识别技术。

生物识别技术(Biometric identification Technology)是利用人体生物特征进行身份认证的一种技术。指纹识别是生物识别的一种。指纹除了其他生物特征都具有的唯一性和稳定性外,它还具有许多独到的优点。首先,指纹样本便于获取,易于开发识别系统,实用性强;其次,一个人的十指指纹皆不相同,这样可以方便地利用多个指纹构成多重口令,提高系统的安全性;第二,指纹识别中使用的模板并非最初的指纹图,而是由指纹图中提取的关键特征,从而模板库对系统存储量的要求较小。现代电子集成制造技术使得我们可以制造相当小的指纹图像读取设备,同时飞速发展的个人计算机运算速度提供了在微机甚至单片机上可以进行两个指纹的比对运算的可能。第三,指纹的形成依赖于胚胎发育时的环境和遗传。世界上几乎没有两个完全相同的指纹。另外,由于匹配算法可靠性也不断提高,指纹识别技术已经非常实用。由于指纹具有稳定性、唯一性以及易于采集的特性,一直以来被认为是最可靠的利用生物特征进行身份鉴定的依据之一。

指纹识别作为模式识别领域中使用最早的,也是最为成熟的生物鉴定技术,有着悠久的历史同时发展也很迅速。苏格兰医生Henry Fauld^[1]于1880年10月28日首次在英国《Nature》上发表论文,指出指纹人各不同,恒久不变被认为是现代指纹识别的起源。1892

年,英国Sir Francis Galton^[1]对指纹进行了系统地研究,并提出了指纹细节特征分类,将指纹分为斗(whorl)、箕(loop)、弧(arch)三大类,标志这指纹分类技术开始受到主流学界的关注。1899年,英国Edward Henry[1]建立了著名的Henry 指纹分类系统并于1901年被英国政府正式采用,随后西方各国亦相继采用,指纹识别应用正式走上了科学化道路。

20世纪60年代中期,美国联邦调查局就要求国家标准局(现该名为国家标准和技术机构)的研究人员使用一定的技术研究一下“读取”人类指纹的可行性。1972^[2]年美国国家标准局给联邦调查局提交了一台原型设备。它的体积相当庞大,但是速度很慢。这台机器不能进行指纹间的比较,只能扫描指纹并将指纹的涡纹转换为实验数据。1975年成功推出第一个商业化系统prinrak250,日本也在同年开始研究,于1982年推出NECAFIS。指纹技术的不断进步使美国联邦调查局最终于1999年建成了第一套综合自动化指纹识别系统,这套设备不仅大大加快了指纹识别的速度,而且提高了指纹比较的自动化水平。911事件后,各国对安全的需求大大增加,而各大厂家也开发出基于指纹识别的产品满足需求,如IBM公司从2005年开始在其THINKPAD笔记本电脑上配置集成式指纹识别器,具备指纹功能的是部分型号的T42系列笔记本电脑,指纹识别器位于电脑的掌托部位,电脑用户只要将手指在一个水平方向的半英寸传感器上滑行,就能实现进入系统、开启应用软件、打开网站、调用数据库等操作。2006年推出的AUDI A8L配备全球第一套指纹智慧辨识系统,可同时完成八种以上的个人化行车设定,包括座椅角度与位置、后视镜的角度、方向盘的位置与高度、空调、音响、MMI多媒体系统,连悬吊系统的软硬/高低设定,以及前后泊车警示系统的音量等,均可藉由这套系统完成所有个人化设定,此外这套系统更可量到家庭的需求,同时可记忆四名驾驶者与20组指纹,避免因受伤影响辨识效能,美国爱国者法案规定,美国进入美国的外国人都要在政府的系统里留下自己的指纹特征。2003年美国国家标准局组织了名为FpVTE2003^[2]的指纹评测,目的是为了评测指纹识别算法的匹配、验证、识别的准确性。随后FpVTE2003成为了入境美国进行指纹验证时的法定指纹评估标准,并成为美国爱国者法案的一部分。

从2000年开始的由美国圣何塞州立大学、密歇根州立大学和意大利博洛尼亚大学等国际权威组织发起的国际指纹识别竞赛(FVC)^[2]是全球生物特征识别技术行业最高级别的竞技大赛,目的是为了全面了解、评价当代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。因此得到了包括我国在内的国际指纹业界、研究机构的普遍关注和认可。目前全球通用的检测方法都是通过国际指纹识别竞赛(FVC)来进行行业测评。FVC同时也成为各国指纹识别技术发展水平的试金石。FVC2004吸引了来自各国的40多个参赛单位。

1.1.2 国内指纹识别的研究应用现状

在指纹自动识别系统开发方面,国外起步较早,而国内对指纹自动识别系统研究较晚。但是我国在指纹识别应用研究方面发展迅速,在指纹识别算法方面在国际上具有比较先进甚至领先的研究水平。中科院自动化所田捷研究员领导的生物特征研究小组开发的指纹识别算法和汉王公司在FVC2004国际指纹识别竞赛中都取得了优异的成绩,在国际同行中很有影响。

但是,在应用技术上特别是硬件采集技术上,国内与国际的水平存在较大差距。以指纹采集的硅芯片为例,国际上的几大厂商几乎垄断了国内所有的市场。而相对容易切入的光学采集仪,高端市场基本也为国际厂商所垄断。同时,应用系统的开发上,主要是应用在门禁系统等小规模的应用。在这一点上与国外有这很大的差距。

1.2 指纹识别的研究方向

指纹识别作为生物识别技术最为成功的应用之一,近年来已得到快速的发展和普及,迄今为止,自动指纹识别技术的研究也已经取得了很大的成绩。包括指纹考勤机、指纹门禁控制器、指纹一体化锁、指纹保险柜等已经被大部数人所接受和喜爱,目前的自动指纹识别系统已具备一定的可靠性、快捷性、灵活性、安全性。但以指纹识别为认证主体的产品或系统目前尚处于起步阶段,具体表现为:

1、指纹图像增强技术有待提高

非接触式真皮层指纹采集生理学的研究表明,指纹的结构在真皮层有着完整和稳定的表现。如果能实现在真皮层采集指纹结构,就可以在指纹采集技术方面实现突破,在指头干、湿、脏和指纹磨损严重的情况下都能准确、清晰、完整地进行指纹采集。通过非接触方式采集指纹,则可以有效解决指纹录入时的变形问题。目前的指纹传感器对被采集指纹的适应性差,在指头较湿、较干、较脏或磨损较为严重时,采集的指纹质量往往不能让人满意。同时指纹采集时的变形问题至今没有得到很好的解决。这样就对指

纹图像增强技术提出了很高的要求，目前存在的指纹图像增强技术经过各国科学家的不断研究已经取得很大的进展，但离理想的效果仍然有很长的距离。

2、指纹分类技术有待突破

对于工作在验证模式下的自动指纹识别系统，指纹分类技术并不是一个问题，但对于工作在辨识模式下的系统，指纹分类技术的研究水平则至关重要。指纹分类技术越完善，能够划分的类型越细，样本数据库每个类别中所包含的样本数量就会越少，对一次识别任务来讲，需要比对的次数和时间开销就会越少。按照现行的分类标准，将指纹分成四类、五类或六类还是远远不够的。现在，较好的指纹分类算法，将指纹分为五类和四类的准确率^[9]分别也只有 87.5% 和 92.3%，而这样的分类结果还远不能满足实际应用的需要。

3、缺乏自动指纹识别系统的性能评价体系。

自动指纹识别系统的性能评价是自动指纹识别领域一个急待解决的问题。严格来讲，自动指纹识别系统的性能评价应该由一个全国乃至世界性的专门检测机构来进行，该机构应该拥有一个样本容量足够大、在质量、类型等方面都具有充分代表性的指纹数据库，制定一套科学、严谨的测试方案，能够对各种自动指纹识别系统的性能给出准确、统一、权威的评价。但是，遗憾的是，迄今为止，国内外还没有这样一个自动指纹识别系统性能评价机构，各个自动指纹识别系统的性能往往是由开发者自己来建立或者选择数据库、设计测试方案、进行性能测试这样，由于各个自动指纹系统在测试时使用的数据库在容量、指纹质量方面各不相同，测试方案差别也比较大，不可避免地造成现在自动指纹识别系统性能评价的混乱和无序，而且各系统间也不存在可比性。所以，尽快建立一个专门的自动指纹识别系统评价机构，对各个自动指纹系统给出准确、统一、权威的测试结果，也是自动指纹识别技术研究领域一项急需解决的问题。

1.3 本论文研究意义和主要内容

1.3.1 本论文的研究意义

自动指纹识别技术是一个学科交叉性很强的研究领域。迄今为止，该技术的研究已经取得了巨大的成就，也面临着一些困难，需要光学、电子、计算机、数学乃至生理等学科的共同参与和努力，才有可能尽快将这一技术完善并实现产业化。短时间内，自动指纹识别产品还不可能在各个领域得到全面的普及和应用，原因主要有3个方面：一是自动

指纹识别技术体系本身还存在一些问题，还需要时间去完善；二是，作为一种新兴事物，一种全新的身份识别手段，人们认可和接受它需要一个过程；三是，自动指纹识别技术的出现，对传统的身份识别手段带来了巨大的冲击，该技术的普及需要结合各应用行业的具体情况，在行业内部做很多配套的改革，而这也需要一个过程。但是毫无疑问，在不久的将来，以指纹为代表的生物识别技术，将会逐步在各行业、各部门得到越来越广泛的应用，并将最终占据身份识别领域的主导地位，成为各行各业和人们生活中不可或缺的一部分，以指纹为代表的生物识别技术具有良好市场前景和巨大市场潜力，已成为大家的共识。更为重要的是，生物识别技术在网络、金融和公共安全等领域的应用，具有重要的战略意义。

1.3.2 本论文的主要内容

本论文主要开展以下方面的研究：

1. 高性能图像预处理算法的改进

本文在很多前人的研究基础上对传统的指纹图像增强算法进行了改进，在实现指纹识别算法思想，不断调整各种参数，最终完成实现高性能算法同时尽量满足国际公认指纹识别算法评估体系标准的软件。改进点主要有：

1) 考虑到指纹的方向和脊频都具有局部一致性，就没必要对图像中的每一点进行统计。本文的做法是将图像分割成大小为的小块，以该小块中心点的方向和脊频作为该小块中每一点的方向和脊频。

2) 对Gabor滤波器的复杂度最大的两个参数(方向和脊频)进行量化，这样就可以用有限个模板满足整个滤波的要求。

2. 具体软件的开发

本论文在具体软件设计中考虑到具体的应用情况，使软件具有良好的可拓展性，在指纹特征数据库设计上并未采用现有的数据库管理软件来实现指纹数据库功能，而是采用 xml 文件来建立数据库，这样做有两个优点：实现基于 web 的认证和便于移植。是考虑到很多指纹识别应用的单位本身就已经采用商用数据库，同时商用的数据库提供 xml 转化成数据库的功能，便于与现有数据库对接。

第二章 指纹识别的原理

指纹因其具有终生不变性及稳定性，而且不同人的指纹特征相同的可能性几乎为零，因此被广泛应用于案例分析、存取款身份鉴别等特别领域。由于指纹识别即使对指纹专家来说也是一项繁琐而枯燥的工作，且识别准确性受专家经验制约，因此人工识别指纹远不能适应实际工作的需要。计算机技术和半导体产业的迅猛发展，为复杂的科学计算和指纹自动识别提供了可能。本章介绍了指纹识别的原理概述了指纹识别领域的最新进展，对每个步骤相关的算法进行总结。另外，我们还阐述并比较指纹图像增强相关的各算法，并描述了最普遍的基于细节点模型的特征提取算法，其次列举指纹图像匹配的各种算法。

2.1 指纹识别理论

自动指纹识别系统的工作模式可以分为两类：验证模式(verification)和辨识模式(identification)。验证模式又称一对一比对(1:1 matching)，比对原理为：指纹预先登记到样本数据库并设定一个标识码，比对时，录入指纹并输入标识码，系统根据标识码从数据库中提取指纹样本与录入指纹进行比对。辨识模式又称一对多比对(1:N matching)，是把录入的指纹与样本数据库中的所有指纹逐一进行对比，直至找到相匹配的指纹或搜索完整个样本数据库后给出无对应指纹的结论。典型的指纹识别系统包括个体指纹采集，图像增强、特征提取、与数据库中模板匹配等步骤，图 2-1 为指纹识别系统的一般结构：

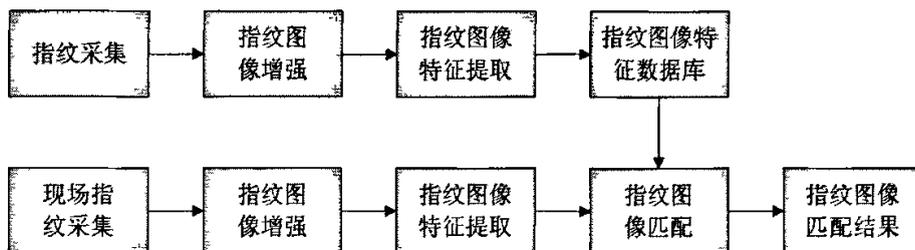


图 2-1 指纹识别系统的一般结构

Fig 2-1 General structure of AFIS system

指纹识别技术的第一步，就是要采集人的指纹，实现这一功能的硬件为指纹传感器。目前市场上指纹传感器都在向着体积更小、成本更低、能力更强的方向发展。指纹传感器根据不同的成像原理，大致有光学式、压感式、热感式和超声式几种类型，而其中已经被广泛应用的是光学式和压感式指纹传感器。指纹传感器采集到的指纹图像的质量好坏，直接影响到后续的图像处理过程。而且，由于指纹采集的应用场合不同，所需要的指纹识别系统的性能也不尽相同，针对不同的应用环境，应该选择合适的指纹传感器，以达到好的识别效果。

指纹传感器是硬件方面的内容，而指纹识别系统除了硬件方面的内容剩下的主要是自动指纹识别软件部分，自动指纹识别软件的核心部分是自动指纹识别算法，一般来讲，自动指纹识别算法体系大致由指纹增强、指纹分类、特征提取和指纹比对四个部分组成。对于工作在验证模式下的自动指纹识别系统，不需要进行指纹分类。图像增强、特征提取和指纹比对则是自动指纹识别系统的软件部分，是核心内容，本课题目前并不涉及指纹分类算法，所以下主要对图像增强、特征提取和指纹比对三方面内容加以重点介绍。

2.2 指纹图像增强

在指纹所有的特征量中，指纹纹线的末梢和分叉细节点是最普遍采用的细节特征结构。所以自动指纹识别系统通常通过比对指纹脊线和谷线结构中的细节特征点来实现个人身份识别。实际上直接从输入指纹图像中准确的提取特征点是十分困难的。这是因为指纹细节点的自动精确提取极大程度上依赖于图像质量。伤疤、与指纹采集仪的不标准接触、采集环境等因素都会导致指纹图像质量的退化。但是质量比较差的指纹经过图像增强后，指纹专家就能够利用诸如局部纹线方向、纹线连续性、纹线趋势和纹线疏密程度等各种视觉线索正确地识别出特征信息。指纹增强技术正是在这样一种背景下产生和发展的。所谓指纹增强，就是对低质量的指纹图像采用一定的算法进行处理，使其纹线结构清晰化，尽量突出和保留固有的特征信息以避免产生伪特征信息，其目的是保证特征信息提取的准确性和可靠性。增强算法的主要目的就是提高指纹纹线的清晰度，削弱图像中的噪声。因此，在特征提取前对指纹图像进行增强是很有必要的，它可以提高计算细节点坐标的可信度，是指纹识别中最重要的步骤之一。使它的结果对后面的识别来说比原始指经纬度图像更合适特征提取以进行识别。只有经过增强后的指纹图像，才能进入到

下一步（指纹的识别预分类）去处理。图像增强结果的好坏直接关系到指纹识别率的高低。图 2-2 为大部分指纹图像增强算法的一般流程，当然不同图像增强算法的具体实现步骤不同：

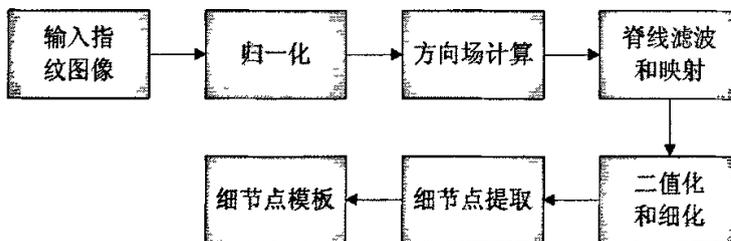


图 2-2 图像增强算法一般流程

Fig 2-2 General Flowchart of Image enhancing Algorithms

指纹图像是通过扫描仪获得模拟信号，并经采样、量化后，以矩阵的形式存入计算机。由于图像的采集为纵列式方式，量化后的指纹图像有许多噪声。在指纹特征中除了模式区等少数异常区域以外，在一个小的局部区域内，指纹纹线的分布具有良好的频率特性和方向特性。充分利用局部区域内纹线的频率和方向信息，对每个局部区域构建相应的模板进行增强就能有效地去除噪声，突出纹线的固有结构。Gabor 函数具有最佳时域和频域连接分辨率的特点，可以很好地兼顾时域和频域信息，实现对信号的处理，所以使用 Gabor 函数实现指纹增强应该是一个合理的选择。

2.2.1 指纹增强算法综述

Daugman^[15]提出了二维 Gabor 滤波器理论，并指出二维 Gabor 滤波器可以同时为空域、频域和方向上获得最佳的分辨率，二维 Gabor 滤波器可以很好的模拟人的视觉神经细胞的感受视野轮廓。Daugman 对人体虹膜研究时，提出一种具有平移不变性和尺度不变性的纹理描述方式，通过对虹膜纹理的多尺度正交 Gabor 小波分解系数的有序排列来描述。实际上，Daugman 所提出的虹膜纹理描述并不真正具备旋转不变性。由于人体头部运动范围的限制，虹膜本身发生较大范围旋转的可能性很小，而较小的旋转可以在匹配阶段通过虹膜编码(IrisCode)本身的旋转来解决。而 Jain^[16]等人成功地使用 Gabor 滤波器组完成纹理分割。由于 Gabor 滤波器具有频率和方向选择的特性，采用 Gabor 滤波器不但能够去掉噪声，并且能把指纹的脊和谷的结构不失真的保留下来。

Anil K Jain^[36]等人利用虹膜识别中 IrisCode 的思想,提出了一种新颖的基于 Gabor 滤波函数库的指纹特征表示法,即 FingerCode。该方法以奇异点为中心将指纹分割为多个小区域,分别用多个方向的 Gabor 滤波器作用于指纹图像,然后用滤波后得到的图像局部区域的平均绝对偏差 AAD 来描述指纹的局部特征。Gabor 滤波算法主要包括以下几个步骤:

(1)求得指纹图像的参考点。

(2)以参考点为中心将指纹图像分成许多个小扇区。

(3)对所选择区域的指纹图像进行归一化。

(4)用 Gabor 滤波器对指纹图像分成 8 个方向(0 度, 22.5 度, 45 度, 67.5 度, 90 度, 112.5 度, 135 度, 157.5 度)进行滤波,得到每个方向的 Gabor 滤波图像。

(5)计算每个 Gabor 滤波图像的平均绝对方差和特征值,得到整个指纹图像的特征表示 FingerCxx: le。

(6)用所求得的指纹图像的特征值 FingerCxx: le 与指纹模板数据库里的模板 FingerCode 特征值进行匹配,计算两者之间的欧拉距离,得到识别结果。

迄今为止,已有不少文献对指纹增强的算法进行了研究^[37]。绝大多数方法都使用了纹线方向作为增强算法的参数,有些方法还同时使用纹线的频率信息。纹线方向和纹线频率都是指纹的固有结构信息,只有充分地使用这些信息才能对指纹图像取得良好的增强效果。

L. Hong 等人^[3]提出了一种使用加博函数增强指纹图像的方法,并对该方法的性能进行了评测。该方法确立了使用 Gabor 函数实现指纹增强的基本技术体系。但可用区域和不可用区域的划分需要大量的典型样本,依靠经验确定参数取值,技术实现的难度比较大;当纹线结构模糊、对比度很低时,基于方向窗的方法难以保证纹线频率计算结果的可靠性。

L. Hong 等人^[4]还利用加博滤波器组将指纹图像分解为一个滤波图像集,用分解方法计算图像方向场的方法。对低质量指纹图像,该方法仍能取得较好的增强效果,但算法相当复杂和耗时,难以满足在线应用的需要。

B. G. Sherlock 等人^[6]使用方向傅里叶滤波器在频域实现了对指纹图像的增强。该算法仅使用了纹线方向信息而没有利用纹线频率信息。D. C. Douglas Hung^[6]专门针对二值图像提出了一种指纹增强和特征纯化的方法。L. O' Gorman 等人^[7]使用的指纹增强方法则高度依赖纹线方向信息。使用 Gabor 图像增强主要包括归一化、方向场计算、脊线和滤波映射、二值化和细化等步骤。下面对使用 Gabor 的指纹图像增强算法一般步骤进行介绍。

2.2.2 指纹图像的分割

在指纹识别的流程中，分割是指纹图像预处理的第 1 步。通过指纹图像分割，可以将指纹图像的前景、背景以及模糊区域区分出来，以使得指纹识别处理的其他步骤在有效的纹线区域进行，从而能够有效地减少干扰引起的伪特征，以提高指纹识别的正确率，并减少运算量。图像分割的目的是将指纹的有效区域从背景和噪声区域中分离出来。有效区域的分割不仅简化了后续的处理，更显著提高了细节特征提取的可靠性。

目前主要有 3 种常用的图像分割方法：基于方差阈值的方法，基于方向场信息的方法和基于频域的方法。

基于方差阈值的图像方法也是最常用的方法，相对于其他区域，有效的指纹区域具有一个较高的方差值，因此，可选取方差阈值作为特征量进行分割。

另一类指纹图像分割算法是基于方向场信息的，它的性能依赖于方向场的可靠性，对灰度对比度不敏感，但实际上在脊线不连续的区域或中心点和奇异点附近的区域中，精确的提取方向场几乎是不可能的。

基于频域的方法^[6]，假定前景及噪声区域内不能保持指纹表面的纹线模式，Fourier 频谱中几乎没有对应的能量分布，但是对指纹弹性形变导致脊线间隔不匀的区域，这种算法尚不能很好地处理。

融合多种特征来实现适当的分割显然是可行的。由于采用了不同的特征，分割处理存在很多不确定的因素，人们提出了一系列的算法来改进这种情况。

2.2.3 图像归一化

指纹图像的采集过程是非理想的，很多因素都可能导致沿脊线和谷线的灰度值出现不同程度的扭曲，譬如墨印的密度不同、与采集仪的不均匀接触等。对图像进行归一化可以减少亮度分布的差异，有利于后面的增强处理步骤。

图像归一化的方法主要是通过将灰度的方差限制到要求范围内^[6]来统一图像的密度值, 指纹图像规一化的主要目的是降低沿脊线、谷线方向的灰度的变化程度, 而不改变脊线和谷线结构的清晰对比度, 使图像具有预定的均值与方差。

2.2.4 计算指纹图像的方向场

指纹图像的方向场都可用一个描绘脊线方向的矢量矩阵表示, 其中每个元素值都是指纹图像对应像素的局部脊线的方向矢量, 它反映了指纹图像的基本信息, 是衡量指纹质量的重要因素。

在诸多方法中, 应用最广泛的方向场计算方法是基于梯度的, 利用方向场矢量垂直于梯度矢量的性质计算指纹图像的方向场, 它将图像分割成小块, 通过计算与块内各像素梯度值垂直的矢量均值估计各小块对应的方向场矢量, 而基于模板对比的方法将方向场量化到有限个方向, 利用特定的模板来估算块内脊线的方向, 与上一种方法相比, 它的特点是计算速度较快但计算结果精度不高#指纹图像中存在噪声及被污染的区域, 因此有必要对已有的方向场进行后处理。由于局部邻域内脊线方向变化缓慢, 用一个低通滤波器进行方向场图像平滑可以有效减少外界的影响。

2.2.5 脊线滤波

由于指纹状况及采集环境是非理想的, 指纹图像常常含有随机噪声和粗糙的脊线。为了精确地绘制指纹, 有必要去除噪声干扰并用滤波器平滑脊线。依据滤波的实现方法不同, 可以将现有的滤波算法大致分为两类: 基于空域的及基于Fourier频域的。前者利用指纹的局部特征包括脊线频率和方向, 而后者主要是基于脊线整体特征。

基于空域的算法通过直接对滤波算子和数字指纹图像求卷积实现带通滤波, 其中有代表性的是参考文献^[6]提出的一种兼有频率选择性和方向选择性的Gabor滤波算法, 但是算法太过耗时而不适合应用, Areekul^[7]提出了一些改进的Gabor滤波算法以加快处理过程。

基于频域的算法通过直接改善原始指纹的频谱来实现滤波处理, 由于脊线和谷线间隔几乎等距, 指纹图像的能量集中在某个频率附近, 参考文献^[9]等提出的方向滤波方法是基于频域的, 首先用若干个方向滤波器分别对原有图像进行处理, 每一个都在提取某个方向频谱的同时削弱其余方向的频谱值, 然后将经方向滤波后的多幅图像融合得到增强后的图像。与Fourier增强相比, 基于空域的方法如Gabor滤波比较复杂而且计算占用资源较多。

2.2.6 二值化和细化

二值图像是指整幅图像画面内仅黑（灰度值为0）白（灰度值为1）二值的图像，在它们上面不呈现出灰度的变化。在数字图像处理中，二值图像占有非常重要的地位。这是因为在实用的图像处理系统中，要求处理的速度高、成本低，信息量大的浓淡图像处理花销太大，不是上策。而且二值化后的图像能够用几何学中的概念进行分析和特征描述，比起灰度图像来说方便得多。因而二值图像处理目前已成为图像处理中的一个独立的、重要分支而获得广泛的应用。二值化主要是根据原始的灰度图像来确定图像上的每一点应属于有效区域还是背景区域，从而产生对应的二值图像。它不仅大大减少存储量，而且对使得后面的判别过程少受干扰，大大简化其后的处理方法。二值化后的图像是以后处理的基础，它的算法对后面的处理有直接的影响。一个好的算法可以得到一个高质量的二值图像。反之，如果该阶段引入噪声，就会直接降低图像质量，影响识别精度。二值化处理将一幅灰度图像转化为二值图像，它提高了指纹图像中脊线和谷线间的对比度，使细节节点的提取变得更方便。二值化的关键问题是选取一个合适的阈值。近年来，普遍采用基于局部适应的阈值的算法来生成二值化图像，它依据每个像素周围的局部图像窗口选择适当的阈值并将该像素判定到前景类或背景类。HE^[9]等利用方向场可以快速而直接的实现图像二值化。

经过二值化后的指纹图像下一步的处理就是指纹图像的细化，由于我们所关心的不是纹线的粗细，而是纹线的有无。因此，在破坏图像连通性的情况下必须去掉多余的信息。在二值图像处理领域，细化是很重要的一个处理环节。如干涉条纹图像，由于条纹粗、宽二边缘弥散，不细化成线状就没有办法去精确地取数计算它。另一方面，一个连接成分如果能用线状结构去值图像的细化问题成为图像处理的一大热门，有不少人纹脊线的轮廓和有关的端点和交叉点。但是，二值化后的脊线有一定的宽度，这种表征它的特性，无论在图像识别还是在数据压缩方面都有重要意义。所以二宽度会给后面的识别增添很多麻烦。因而应先将指纹脊线的宽度采用逐渐剥离的方法，使得脊线成为只有一个像素宽的细线，这将非常有利于下一步的分析，这个过程叫细化。其目的是用一组细线来刻画一个连接成分，这不仅能达到压缩数据量的目的，而且易于对连接成分的特征提取。因而，这一操作过程必然需要删除某些像素。

指纹图像细化后，还可以根据指纹得到一般处理规律。进行后加工处理，如去掉无效

的纹线搭桥及连接豁口等。还可以采用人机交互方式，对残缺指纹进行人工修补。作为细节提取前的最后一个步骤，细化是一种基于形态学的处理方法，它持续腐蚀原有的脊线到一个像素宽为止。标准的细化算法^[9]是通过重复执行两次子细化实现的。

2.3 细节提取

指纹纹路并不是连续的，平滑笔直的，而是经常出现中断、分叉或打折。这些断点、分叉点和转折点就称为“节点”。就是这些节点提供了指纹唯一性的确认信息。指纹其实是比较复杂的。多年来在各个公司及其研究机构产生了许多数字化的算法。但指纹识别算法最终都归结为在指纹图象上找到并比对指纹的特征。

指纹的特征。我们定义了指纹的两类特征来进行指纹的验证：总体特征和局部特征。

2.3.1 指纹的全局特征

总体特征是指那些用人眼直接就可以观察到的特征。指纹的全局特征主要是纹路图案、模式三角点、核心点、式样线、纹数等。

基本纹路图案。指纹的纹路图案可被肉眼观察得知，主要根据他们图案的形状分类。最基本的三种纹路图案是环型（loop），弓型（arch），螺旋型（whorl）。其他的指纹图案都基于这三种基本图案。仅仅依靠图案类型来分辨指纹是远远不够的，这只是一个粗略的分类，但通过分类使得在大数据库中搜寻指纹更为方便，这也是指纹分类算法要研究的内容。图 2-3 为基本指纹纹路图案：



图 2-3 三种基本指纹纹路图案

Fig 2-3 Three basic grain pattern

模式区 (Pattern Area)

模式区是指指纹上包括了总体特征的区域，即从模式区就能够分辨出指纹是属于那一种类型的。有的指纹识别算法只使用模式区的数据。指纹识别算法应该所取得的完整指纹而不仅仅是模式区进行分析和识别。

核心点 (Core Point)：核心点位于指纹纹路的渐进中心，它用于读取指纹和比对指纹时的参考点。

三角点 (Delta)：三角点位于从核心点开始的第一个分叉点或者断点、或者两条纹路会聚处、孤立点、折转处，或者指向这些奇异点。三角点提供了指纹纹路的计数和跟踪的开始之处。

式样线 (Type Lines)：式样线是在指包围模式区的纹路线开始平行的地方所出现的交叉纹路，式样线通常很短就中断了，但它的外侧线开始连续延伸。

纹数 (Ridge Count)：指模式区内指纹纹路的数量。在计算指纹的纹数时，一般先在连接核心点和三角点，这条连线与指纹纹路相交的数量即可认为是指纹的纹数。图 2-4 为指纹模式区：

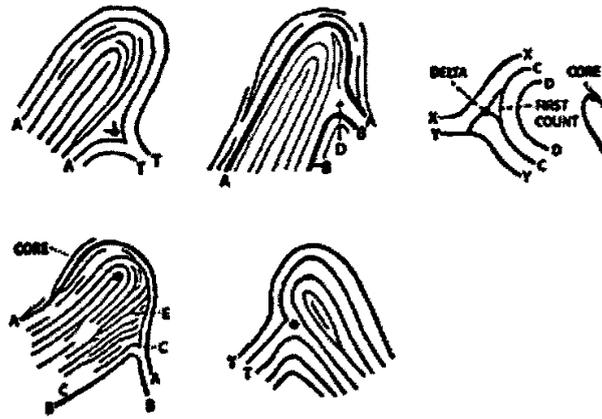


图 2-4 指纹模式区

Fig 2-4 fingerprint pattern zone

2.3.2 指纹的局部特征

局部特征是指纹上的节点。两枚指纹经常会具有相同的总体特征，但它们的局部特征——节点，却不可能完全相同。指纹上的节点有四种不同特性：

1. 分类 - 节点有以下几种类型，最典型的是终结点和分叉点

A. 终结点 (Ending) — 一条纹路在此终结。

B. 分叉点 (Bifurcation) — 一条纹路在此分开成为两条或更多的纹路。

C. 分歧点 (Ridge Divergence) — 两条平行的纹路在此分开。

D. 孤立点 (Dot or Island) — 一条特别短的纹路，以至于成为一点

E. 环点 (Enclosure) — 一条纹路分开成为两条之后，立即有合并成为一条，这样形成的一个小环称为环点

F. 短纹 (Short Ridge) — 一端较短但不至于成为一点的纹路，

图 2-5 为常见的指纹分类：

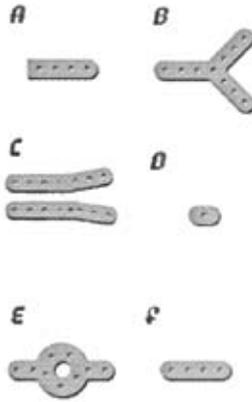


图 2-5 常见的指纹分类

Fig 2-5 General types of minutiae

2. 方向 (Orientation) — 节点可以朝着一定的方向。图 2-6 为指纹方向场:

2		3		4		5		6
1		2	3	4	5	6		7
		1				7		
0		0		*		0		0
		7				1		
		6	5	4	3	2		1
7								
6		5		4		3		2

图 2-6 指纹方向场

Fig 2-6 Fingerprint direction

3. 曲率 (Curvature) — 描述纹路方向改变的速度。

4. 位置 (Position) — 节点的位置通过 (x, y) 坐标来描述, 可以是绝对的, 也可以是相对于三角点或特征点的。

指纹的细节特征(minutiae)可以有 150 种之多. 但这些特征出现的概率并不相等,

很多特征是极其罕见的，一般在自动指纹识别技术中只使用两种细节特征：纹线端点(ridge ending)和分叉点(ridge bifurcation)。纹线端点指的是纹线突然结束的位置，而纹线分叉点则是纹线突然一分为二的位置。大量统计结果和实际应用证明，这两类特征点在指纹中出现的次数最多、最稳定，而且比较容易获取。更重要的是，使用这两类特征点足以描述指纹的唯一性。通过算法检测指纹中这两类特征点的数量以及每个特征点的类型、位置和所在区域的纹线方向是特征提取算法的任务。

细节特征的坐标直接表征该指纹，因此特征提取的好坏直接影响到以后的指纹匹配的结果，所以特征提取是指纹识别系统的关键部分之一。如果输入图像的质量很好，很容易确定其结构取细节节点的简单过程。具备很好的脊线结构，但实际上，此时的特征提取只是从细化后的脊映射使得特征提取的准确性受到影响。在进行特征提取的过程中一般要进行细节点处理。一般在提取细节点之前指纹识别系统研究我们对细化后的指纹图像进行分支和岛屿的去除，这样可由于受很多因素的影响，输入指纹图像并不以大量减少伪端点和伪分叉点的数量。尽量去除指纹图像边缘的细节点，尤其是图像边缘提取出的端点，几乎 100%为伪端点。细节点提取后还要进行伪细节点去除，不过目前大多数算法都是经验算法，如相邻细节点距离小于一定像素即认为其为伪细节点。对最终检测出来的每一个细节点，我们一般记录如下信息：

1. 细节点的坐标(x, y)。
2. 细节点的方向，这个方向定义为该细节点所在的局部脊线的方向。
3. 细节点的类型，即端点或分叉点。
4. 细节点对应的脊线。

有些系统记录更多的信息，如细节点的相互位置关系、细节点到其他细节点的距离、细节点到图像中心点的距离、细节点之间的脊线数等。这样，就将一幅指纹图像转化成了一个由细节点组成的平面点集。

提取过程可能存在错误，导致丢失真实细节点或错误生成伪细节点。因此选择适当的后处理以提高细节点的可信度是很有必要的。Luo^[1]等利用人们在指纹方面的先验知识来处理脊线断裂、搭桥、污损及伤疤等情况，得到更可靠的细节点。Chikkerur^[2]等提出的方法主要是基于下述规则：

- (1)合并相互间距离小于设定阈值且方向相似的细节点;
- (2)除去指纹图像中感兴趣区域边界的所有细节点;
- (3)丢弃那些方向和局部脊线方向不一致的细节点等等,然后我们得到输入指纹图像的细节点模式。

2.4 指纹图像匹配算法

目前存在的指纹识别算法主要包含以下几种:包括基于点模式的匹配、特征转换、结构匹配及基于图的匹配。本课题实现了基于点模式的二叉树匹配,下面主要介绍基于点模式的匹配。

大多数指纹识别算法是基于细节点匹配的,因为普遍认为细节点是最独特、最可信的特征^[20],基于点模式的匹配算法假定输入指纹及模板指纹间的存在平移、旋转、尺寸缩放等相似变换,而且它们可适应限定范围内的伪细节点出现和真细节点的丢失,其中一些改进后的算法^[23]可容忍细节点在一个较小距离内的偏移。

点模式匹配的问题是模式识别中的一个有名的难题。它是指对于两个含有不同数量的点集 $P\{p_0, p_1, \dots, p_m\}$ 和 $Q\{q_0, q_1, \dots, q_n\}$, 如何找出它们之间的匹配关系。因此,一个好的点模式匹配算法应该能够有效地解决两个点集间的几何不变量问题。人们对一般的点模式匹配提出过很多算法。Ranade 等人的松弛算法, Skea 的三角形匹配算法等,这些算法有的只是针对部分几何不变量进行处理,如松弛算法主要处理点集间的平移,有的计算复杂度太高,如三角形法达到了 $O(N^2)$ 。

点模式匹配中的 2 个点集 P 和 Q , 其中 P 从第 1 幅图象抽取, 由 m 个点特征构成, Q 从第 2 幅图象抽取, 由 n 个点特征构成, 即 $P = \{p_0, p_1, \dots, p_m\}$ 和 $Q = \{q_0, q_1, \dots, q_n\}$ 。它们之间的匹配就是找到一个校准函数 $G(t_x, t_y, s, \theta)$, 以使两个点集有最大数量点对之间存在稳定的一一对应关系。即 $G(P) = Q$, 而在实际应用中, 点的相对位置存在误差, $G(P)$ 不可能完全等于 Q , 所以我们一般用 $\|G(p_i) - q_j\| \leq d$, d 为允许的变形范围)来表示 Q 与 P 中点对的对应关系。

校准函数 $G(t_x, t_y, s, \theta)$ 是一个含有 4 个参数的函数, 其中 s 是两幅图象间的伸缩系数, 是偏转角度, t_x 和 t_y 分别是沿 X 方向和 Y 方向的位移。点匹配就是对于 2 个从指纹图象

第二章 指纹识别的原理

中抽取的细节特征点集，找出它们间的校准函数 $G(t_s, t_p, s, \theta)$ 。

自动指纹识别系统中，提取后的指纹节点一般用一个四维向量 (x, y, β, c) 表示。其中 x, y 分别是该细节点的坐标位置， β 是该细节点的方向（该细节点所在局部脊线的方向）， c 为细节点的类型。

下面介绍一种常用的四叉树（quadtrees）指纹节点匹配算法，这种算法的基本思想是这样的：判别一种特征是否存在于图像的某个位置，通常方式如图 2-7 所示：

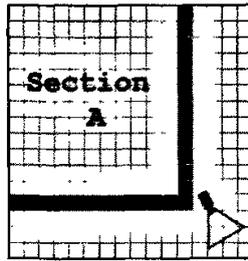


图 2-7 判断特征是否在图内

Fig 2-7 Judge whether the characteristics in the image

quadtrees 是从图像中获得的，把它分割成四个较小的部分，每一个部分继续分下去，直到一个分到一个设定的大小，如图 2-8 所示，首先将他分为四份，然后每一分再继续划分。

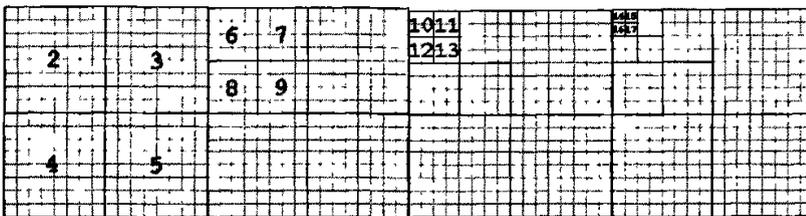


图 2-8 图像的分割

Fig 2-8 Image Segmentation

想上面这样就生成了一个四叉树，这个树是父子关系，每个子节点有一个父节点，每个父节点有四个子节点。叶节点是允许的最小的子节点，每个树都有一个根节点，它没有父节点，但有四个子节点。

在四中树查找相应的特征时，从根节点开始，从根节点开始检测特征是否在根节点的包围坐标内。如果在根节点内，就可以判定特征在根节点的一个子节点内，接着测试子节点。检测到特征不在 2 节点内，图 2-9 为特征查找：

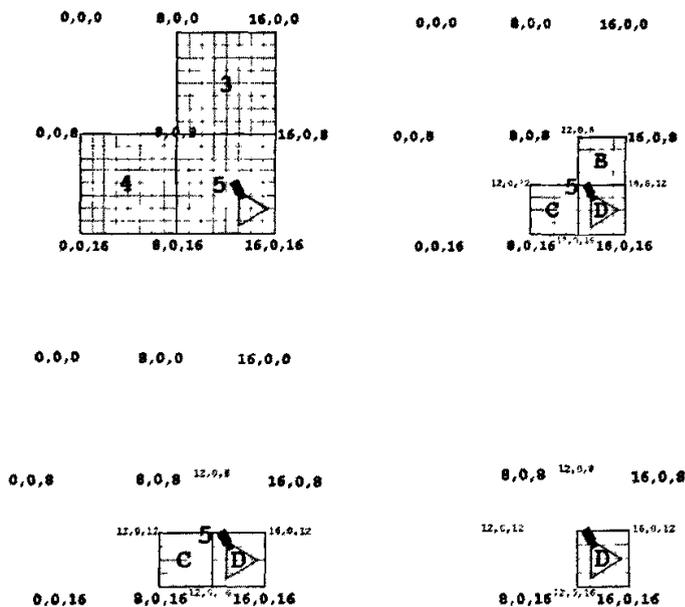


图 2-9 特征的查找

Fig 2-9 Search of the characteristic

然后分别测试节点 3、4 和 5，发现特征在节点 5 里面，接着往下寻找最终在 5 的子节点寻找到了特征的坐标位置。指纹匹配的四叉树匹配算法就是这种原理，下一章具体实现了四叉树的代码实现。

第三章 指纹识别软件的设计

3.1 软件介绍

3.1.1 系统的设计结构

为了把指纹识别系统各个部分的算法整合起来，完成一个切实可行的指纹识别软件，本软件是在 windows xp sp2 环境下开发，应用的编译软件是 MS Visual C++ 6.0。下面为程序的主要框架类：

- 1) `CEFingerApp` 类：程序主线程类，负责程序的启动、关闭消息泵的运用。
- 2) `CMainFrame` 类：程序的主窗口类，负责创建主界面，包括菜单、工具条、以及其他视图。
- 3) `CEFingerDoc` 类：核心文档类，主要负责文档的管理。
- 4) `CEFingerView` 类：输入和控制视图类，用来输入各种指纹算法的参数，控制指纹算法的运行以及视图的显示。
- 5) `CMarkup` 类：xml 管理类，包括节点的添加、删除、查找与更新。指纹特征数据的存储、更新、删除和新建。
- 6) `CAddDlg`, `CfingDlg`, `CEditDlg`：显示视图类，用来显示指定的指纹图像和数据信息。

图 3-1 为程序框架类各类之间的关系：

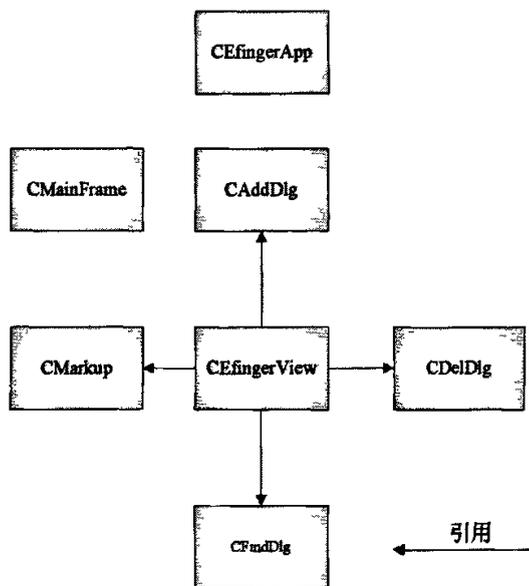


图 3-1 主要程序框架类

Fig 3-1 Main Framework class of program

3.1.2 算法实现函数

图 3-2 为算法实现的各个函数：

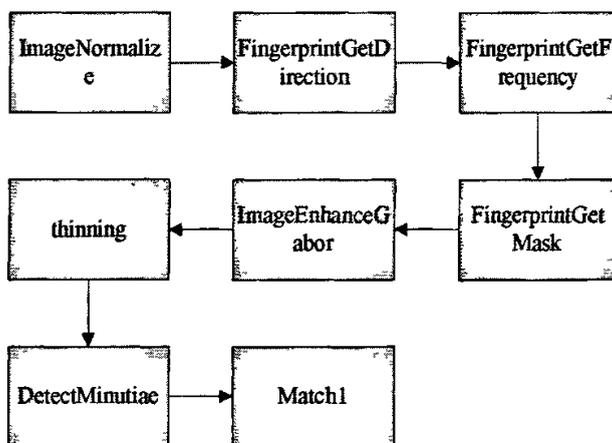


图3-2 算法实现函数

Fig 3-2 Functions realizing the algorithms

这一部分主要包括指纹的处理、指纹匹配等各个环节的函数及数据结构的定义。其中指纹处理中包括了ImageNormalize(图像归一化)、FingerprintGetDirection(方向场计算)、FingerprintGetFrequency(脊频计算)、FingerprintGetMask(计算gabor模板)、ImageEnhanceGabor(图像增强)、Thinning(图像二值化和细化)、DetectMinutiae(节点提取)、Match1(节点匹配)。

运行该软件后,图3-3为软件的界面,3-4为增强处理的对话框:

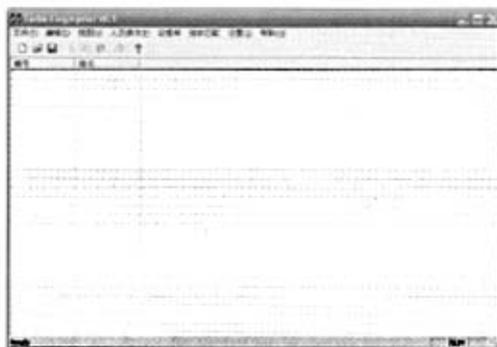


图 3-3 软件主界面

Fig 3-3 Interface of the software



图 3-4 图像增强对话框

Fig 3-4 Image enhancing dialog

3.2 指纹图像的预处理

Gabor滤波的本质是提取图像的特征分量,通常应用的领域有指纹识别,虹膜识别,人脸识别等。有关进一步的原理和技术实现的细节我们将在下面中继续讨论。

3.2.1 图像归一化

指纹图像规一化的主要目的是降低沿脊线、谷线方向的灰度的变化程度,而不改变脊线和谷线结构的清晰对比度,使图像具有预定的均值与方差。在本文所述的算法中,若不进行规一化,Gabor滤波后会产生大量的饱和区。下面算法具体实现的核心片段:

```
/* 计算均值和方差 */
fmean  = (FvsFloat_t)HistogramGetMean(histogram);
fsigma = sqrt((FvsFloat_t)HistogramGetVariance(histogram));
fmean0 = (FvsFloat_t)mean;
fsigma0 = sqrt((FvsFloat_t)variance);
if (fsigma>0.0){
    fcoeff = fsigma0/fsigma;
    for (y = 0; y < h; y++){
        for (x = 0; x < w; x++)
        {
            fgray = (FvsFloat_t)P(x,y);
            fgray = fmean0 + fcoeff*(fgray - mean);
            if (fgray < 0.0)    fgray = 0.0;
            if (fgray > 255.0) fgray = 255.0;
            P(x,y)= (uint8_t)fgray;
        }
    }
}
```

归一化以获得预定的均值和方差。图3-5为归一化后的图像:



图 3-5 归一化后的图像

Fig 3-5 Image after Normalization

3.2.2 图像分割和计算区域方向

考虑到指纹的方向和脊频都具有局部一致性，就没必要对图像中的每一点进行统计。本文的做法是将图像分割成大小为的小块，以该小块中心点的方向和脊频作为该小块中每一点的方向和脊频。区域的方向是Gabor滤波器中的最重要的参数，如果滤波器使用的方向与实际的方向相差的角度太大，滤波的效果会适得其反。在边沿检测中，常用的一种模板是Sobel算子。Sobel算子有两个，一个是检测水平边沿的；另一个是检测垂直边沿的。Sobel算子对于象素的位置的影响做了加权，因此效果更好。Sobel算子另一种形式是各向同性Sobel(Isotropic Sobel)算子，也有两个，一个是检测水平边沿的，另一个是检测垂直边沿的。各向同性Sobel算子和普通Sobel算子相比，它的位置加权系数更为准确，在检测不同方向的边沿时梯度的幅度一致。由于Sobel算子是滤波算子的形式，用于提取边缘，可以利用快速卷积函数，简单有效，因此应用广泛。美中不足的是，Sobel算子并没有将图像的主体与背景严格地区分出来，换言之就是Sobel算子没有基于图像灰度进行处理，由于Sobel算子没有严格地模拟人的视觉生理特征，所以提取的图像轮廓有时并不能令人满意。在观测一幅图像的时候，我们往往首先注意的是图像与背景不同的部分，正是这个部分将主体突出显示，基于该理论，我们给出了下面阈值化轮廓提取算法，该算法已在数学上证明当像素点满足正态分布时所求解是最优的。用Sobel算子计算方向能够很好的满足Gabor的需要。如直接使用的窗口，对于干、湿非常严重的指纹极易求错方向，根据Sobel的特性窗口越大，所求得的方向越准确，以所求块的中心点为中心，选取大小为 $W \times W$ 的块(以大块的方向来估计小块的方向)，本文采用的 W 值为16。具体算法如下：

(1) 采用Sobel算子^[17] 计算每一像素点*(i, j)*的*G_x(u, v)* (沿*x*轴方向)和*G_y(u, v)* (沿*Y*轴方向)其模板系数如公式3-1:

$$\begin{matrix} \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} & \begin{pmatrix} 1 & 2 & -1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \\ \text{x轴方向} & \text{Y轴方向} \end{matrix} \quad (3.1)$$

(2) 计算每块以点*(i, j)*为中心的局部方向 $\theta(i, j)$ ，公式见3-2, 3-3, 3-4:

$$V_x(i, j) = \sum_{u=i-w/2}^{i+w/2} \sum_{v=j-w/2}^{j+w/2} 2G_x(u, v)G_y(u, v) \quad (3.2)$$

$$V_y(i, j) = \sum_{u=i-w/2}^{i+w/2} \sum_{v=j-w/2}^{j+w/2} (G_x^2(u, v) - G_y^2(u, v)) \quad (3.3)$$

$$\theta(i, j) = \frac{1}{2} \arctan\left(\frac{V_y(i, j)}{V_x(i, j)}\right) \quad (3.4)$$

$\theta(i, j)$ 为指纹的纹线方向。由于反正切函数的值域为到，故的 $\theta(i, j)$ 取值范围为到，所以判断 $\theta(i, j)$ 所在象限应该根据 $V_x(i, j)$ 和 $V_y(i, j)$ 的值。若 $\theta(i, j)$ 在第二或第三象限，则求出的 $\theta(i, j)$ 应相应地加上。当 $V_x(i, j)$ 和 $V_y(i, j)$ 同时为零时，就无法得到 $\theta(i, j)$ ，在这种情况下，应该把 $\theta(i, j)$ 设为某个默认数值(本文取0)，否则将会对后续处理带来难以预料的影响。

将 $\theta(i, j)$ 进行量化，实现算法步骤如下：

1) 将弧度值 $\theta(i, j)$ ，转换为角度值 $\theta'(i, j)$ 。

2) 求方向量化后的序号。方法如下（见公式3-5）：

$$xuhao = (\text{int})\left(\frac{\theta'(i, j) \times 8}{180} + 0.5\right) \quad (3.5)$$

式中int表示取整，xuhao的取值范围为：0, 1, 2, 3, 4, 5, 6, 7, 8。分别代表角度：0, 22.5, 45, 67.5, 90, 112.5, 135, 157.5, 180。

```

/* 1 - 把图像分割成块 */
for (y = nBlockSize+1; y < h-nBlockSize-1; y++)
for (x = nBlockSize+1; x < w-nBlockSize-1; x++)
{
    /* 2 - 计算每一块的中心点的梯度 */
    for (j = 0; j < (nBlockSize*2+1); j++)
    for (i = 0; i < (nBlockSize*2+1); i++)
    {
        dx[i][j] = (FvsInt_t)
                    (P(x+i-nBlockSize, y+j-nBlockSize) -
                     P(x+i-nBlockSize-1, y+j-nBlockSize));
        dy[i][j] = (FvsInt_t)
                    (P(x+i-nBlockSize, y+j-nBlockSize) -
                     P(x+i-nBlockSize, y+j-nBlockSize-1));
    }

    /* 3 - 计算方位 */
    nx = 0.0;
    ny = 0.0;
    for (v = 0; v < (nBlockSize*2+1); v++)
    {
        for (u = 0; u < (nBlockSize*2+1); u++)
        {
            nx += 2 * dx[u][v] * dy[u][v];
            ny += dx[u][v]*dx[u][v] - dy[u][v]*dy[u][v];
        }
    }

    /* 计算角度 (-pi/2 .. pi/2) */

```

```

if (nFilterSize>0)
    theta[x+y*w] = atan2(nx, ny);
else
    out[x+y*w] = atan2(nx, ny)*0.5;

```

3.2.3 脊频的检测

其形式如下见公式 3-6,3-7,3-8:

$$x(k) = \frac{1}{w} \sum_{d=0}^{w-1} G(u, v) \quad k=0,1,2,\dots,l \quad (3.6)$$

$$u = i + (d - \frac{w}{2}) \cos \theta'(i, j) + (k - l/2) \sin \theta'(i, j) \quad (3.7)$$

$$v = j + (d - \frac{w}{2}) \sin \theta'(i, j) - (k - l/2) \cos \theta'(i, j) \quad (3.8)$$

具体算法的实现:

```

/* 1 - 首先把图像分割成(16 x 16)的子块 */
for (y = BLOCK_L2; y < h-BLOCK_L2; y++)
{
    for (x = BLOCK_L2; x < w-BLOCK_L2; x++)
    {
        /* 2 - oriented window of size l x w (32 x 16) in the ridge dir */
        dir = orientation[(x+BLOCK_W2) + (y+BLOCK_W2)*w];
        cosdir = -sin(dir); /* ever > 0 */
        sindir = cos(dir); /* -1 ... 1 */

        /* 3 - compute the x-signature X[0], X[1], ... X[l-1] */
        for (k = 0; k < BLOCK_L; k++)
        {

```

```

Xsig[k] = 0.0;
for (d = 0; d < BLOCK_W; d++)
{
    u = (FvsInt_t) (x + (d-BLOCK_W2)*cosdir + (k-BLOCK_L2)*sindir);
    v = (FvsInt_t) (y + (d-BLOCK_W2)*sindir - (k-BLOCK_L2)*cosdir);
    /* clipping */
    if (u<0) u = 0; else if (u>w-1) u = w-1;
    if (v<0) v = 0; else if (v>h-1) v = h-1;
    Xsig[k] += p[u + (v*pitchi)];
}
Xsig[k] /= BLOCK_W;
}

/* Let T(i, j) be the avg number of pixels between 2 peaks */
/* find peaks in the x signature */
peak_cnt = 0;
/* test if the max - min or peak to peak value too small is,
then we ignore this point */
pmax = pmin = Xsig[0];
for (k = 1; k < BLOCK_L; k++)
{
    if (pmin>Xsig[k]) pmin = Xsig[k];
    if (pmax<Xsig[k]) pmax = Xsig[k];
}
if ((pmax - pmin)>64.0)
{
    for (k = 1; k < BLOCK_L-1; k++)
        if ((Xsig[k-1] < Xsig[k]) && (Xsig[k] >= Xsig[k+1]))

```

```

        {
            peak_pos[peak_cnt++] = k;
        }
    }
    /* compute mean value */
    peak_freq = 0.0;
    if (peak_cnt >= 2)
    {
        for (k = 0; k < peak_cnt-1; k++)
            peak_freq += peak_pos[k+1]-peak_pos[k];
        peak_freq /= peak_cnt-1;
    }
    /* 4 - must lie in a certain range [1/3, 1/25] */
    if (peak_freq > 25.0)
        out[x+y*w] = 0.0;
    else if (peak_freq < 3.0)
        out[x+y*w] = 0.0;
    else
        out[x+y*w] = 1.0/peak_freq;
    }
    /* 5 - interpolated ridge period for the unknown points */
    /*
    for (y = BLOCK_L2; y < h-BLOCK_L2; y++)
    {
        for (x = BLOCK_L2; x < w-BLOCK_L2; x++)
        {
            if (out[x+y*w] < EPSILON)
            {

```

```

if (out[x+(y-1)*w]>EPSILON)
{
    out[x+(y*w)] = out[x+(y-1)*w];
}
else
{
    if (out[x-1+(y*w)]>EPSILON)
        out[x+(y*w)] = out[x-1+(y*w)];
}
}
}
*/
/* 6 - Inter-ridges distance change slowly in a local neighbourhood */
for (y = BLOCK_L2; y < h-BLOCK_L2; y++)
{
    for (x = BLOCK_L2; x < w-BLOCK_L2; x++)
    {
        k = x + y*w;
        peak_freq = 0.0;
        for (v = -LPSIZE; v <= LPSIZE; v++)
        {
            for (u = -LPSIZE; u <= LPSIZE; u++)
            {
                peak_freq += out[(x+u)+(y+v)*w];
                freq[k] = peak_freq*LPIFACTOR;
            }
        }
        /*
        freq[k] = out[k];*/
    }
}

```

}

}

3.2.3 计算 Gabor 模板

其形式如下^[16]:

$$h(x, y, \theta, f) = \exp\left\{-\frac{1}{2}\left(\frac{(X)^2}{\delta_x^2} + \frac{(Y)^2}{\delta_y^2}\right)\right\} \cos(2\pi fX) \quad (3.9)$$

$$X = x \cos(\theta) \quad (3.10)$$

$$Y = y \sin(\theta) \quad (3.11)$$

式中 x, y 为求得的模板内的坐标值, θ 为区域内沿指纹方向的角度 f 为区域内脊频(纹线间距的倒数), δ_x 和 δ_y 为高斯包络层的标准偏差。

脊频是Gabor滤波的基础。通常的做法如下:

- 1) 沿指纹方向选取一个比较大的窗口。
- 2) 统计该窗口内谷线的条数。
- 3) 根据统计结果计算谷线之间的平均间距。

以上做法不仅需要进行坐标变化, 而且窗口内必须有三条以上的谷线, 否则, 计算结果误差很大。对于一幅质量较好的指纹, 由于噪声的影响, 计算出的纹线间距的误差在1~3像素之间; 对于较干和较湿的指纹, 计算的结果往往和实际的值相差很大。经实验^[27]发现, Gabor滤波器对纹线间距的准确性并不敏感, 即当Gabor滤波器中使用的纹线间距与实际的纹线间距的误差在一定范围内时, 不会影响处理效果, 实际上不同人的指纹对于同一指纹传感器来说, 指纹的纹线间距的差别满足这个误差范围。

根据以上所述, 对于同一指纹传感器踩得的指纹进行Gabor滤波时, 完全可以使用固定的纹线间距, 这样省去求脊频(即纹线间距的倒数), 大大减少的运算量, 提高了效率, 同时也使过分干湿的区域(无法正确求取纹线间距的区域)有个比较好的处理效果。在进行改进前的Gabor滤波时, 对于不同的区域都需要求取一个滤波模板 $h(\theta, f)$ 。对一幅大小

为 300×300 的图像进行处理时,用(5)运算的次数一定大于 $13 \times 13 \times 13$ 。

在进行求模板运算时,对于该区域只有 f, θ 两个变量,令模板为 $h(\theta, f)$ 。从前面的讨论可以知道 f 和 θ 一共只有9种组合,即共有9种 $h(\theta, f)$ 模板,这样在滤波时可以先将所用到的9种 $h(\theta, f)$ 求出来,然后按照所求的量化的xuhao值(4)以索引方式去调用 $h(\theta, f)$,这样可以大大的减少处理时间。

```
/* width & height of the input image */
FvsInt_t w      = ImageGetWidth (image);
FvsInt_t h      = ImageGetHeight(image);
FvsByte_t* out;
FvsInt_t pitchout;
FvsInt_t pos, posout, x, y;
FvsFloat_t* freq = FloatFieldGetBuffer(frequency);

if (freq==NULL)
    return FvsMemory;

/* TODO: add sanity checks for the direction and mask */
nRet = ImageSetSize(mask, w, h);
if (nRet==FvsOK)
    nRet = ImageClear(mask);
out = ImageGetBuffer(mask);
if (out==NULL)
    return FvsMemory;
if (nRet==FvsOK)
{
    pitchout = ImageGetPitch(mask);
```

```

for (y = 0; y < h; y++)
    for (x = 0; x < w; x++)
    {
        pos    = x + y * w;
        posout = x + y * pitchout;
        out[posout] = 0;
        if (freq[pos] >= freqmin && freq[pos] <= freqmax)
        {
/*            out[posout] = (uint8_t)(10.0/freq[pos]);*/
            out[posout] = 255;
        }
    }
/* fill in the holes */
for (y = 0; y < 4; y++)
    (void) ImageDilate(mask);
/* remove borders */
for (y = 0; y < 12; y++)
    (void) ImageErode(mask);
}
return nRet;

```

3.2.4 Gabor 滤波

利用上面得到的Gabor模板，对指纹图像每个区域进行滤波，见公式3.12：

$$f'(i, j) = \sum_{m=-N_w/2}^{N_w/2} \sum_{n=-N_w/2}^{N_w/2} h(m, n) f(i-m, j-n) \quad (3.12)$$

式中 $f(i, j)$ 和 $f'(i, j)$ 分别为滤波前后的图像， n_w 是滤波器窗口的大小(本文取 10)。

本文对改进前和改进后的处理流程作了比较：

改进前的处理流程：

1) 图像规一化；

- 2) 计算局部区域的方向;
- 3) 计算局部区域的脊频;
- 4) 计算局部区域的 Gabor 滤波模板;
- 5) 进行卷积计算, 计算滤波结果。

改进后的处理流程:

- 1) 图像规一化;
- 2) 计算量化的 9 个滤波模板;
- 3) 计算局部区域的方向;
- 4) 通过量化的 xuhao 索引滤波模板;
- 5) 进行卷积计算, 计算滤波结果。

```
FvsInt_t pitchE = ImageGetPitch (enhanced);
FvsByte_t* pE   = ImageGetBuffer (enhanced);
if (pE==NULL)
    return FvsMemory;

(void) ImageClear (enhanced);
for (j = Wg2; j < h-Wg2; j++)
for (i = Wg2; i < w-Wg2; i++)
{
    if (ImageGetPixel(mask, i, j)!=0)
    {
        sum = 0.0;

        o = orientation[i+j*w];
        f = frequence[i+j*w];

        for (v = -Wg2; v <= Wg2; v++)
```

```

for (u = -Wg2; u <= Wg2; u++)
{
    sum+=EnhanceGabor((FvsFloat_t)u, (FvsFloat_t)v, o, f)*
        pG[(i-u)+(j-v)*pitchG];
}

/* printf("%6.1f ", sum);*/

if (sum>255.0) sum = 255.0;
if (sum<0.0) sum = 0.0;

pE[i+j*pitchE] = (uint8_t)sum;

```

下图为指纹图像增强后的图像:



图 3-6 增强后的图像

Fig 3-6 Image after enhancement

3.2.5 二值化和细化

得到图像每个像素处的方向场后,再依据方向场来对图像进行二值化。若该像素处的脊线方向为*i*,先用估计方向场时用到的方法计算该像素处在方向*i*和垂直方向*iVer*=(*i*+4) mod 8的灰度平均值*Gmean*[*i*]和*Gmean*[*iVer*],然后将该像素二值化为:

$$iVer = \begin{cases} 255 & Gmean[i] \geq Gmean[iVer] \\ 0 & Gmean[i] < Gmean[iVer] \end{cases} \quad (3.13)$$

其中*iVer*表示二值图像中该像素处的值,255为二值图像中图像背景和谷线的灰度值,0为二值图像中图像脊线的灰度值。

由于我们所关心的不是纹线的粗细,而是纹线的有无。因此,在不破坏图像连通性的

情况下必须去掉多余的信息。细化后的图像是用一组单像素宽度的细线来刻画一个连接成分，这不仅能达到压缩数据量的目的，而且易于对连接成分的特征提取，本文采用一个一次通过的并行细化算法^[4]，算法中的各模板针对一般细化算法的缺陷进行了针对性的改进。图中，“c”为当前点。“p”或“q”为“1”，将各模板旋转 90° 的整数倍，则共得到 20 个模板，能对目标的上、下、左、右的轮廓点进行删除。图 3-7^[9]为细化模板：

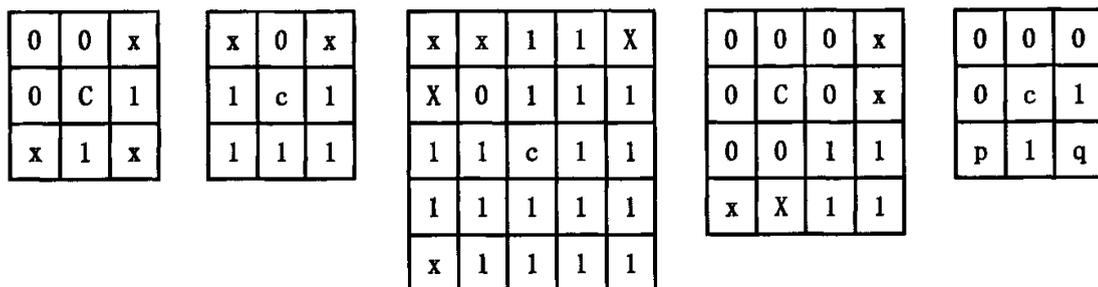


图 3-7 细化模板

Fig 3-7 Thinning mask

模板(a)删除目标拐弯处外缘的轮廓点；模板(b)的作用是删除 4-近邻的边界点，并不破坏骨架的 8-连接性；模板(c)删除目标拐弯处内边缘的轮廓点，其作用是保持拐弯的尖锐，克服了其他细化算法使拐弯圆滑畸变的缺陷。细化算法普遍存在把四交叉点细化成三交叉点和把“T”型结构细化成“Y”型结构的畸变，模板(c)在很大程度上抑制了这种畸变。另外，由于模板(c)的使用，使当前点周围的几个内部点变成了轮廓点，从而加快了细化速度模板；(d)和(e)的作用是抑制由单象索引引起的噪声骨架的生长，在一定程度上克服了细化产生伪线段的缺陷。细化模板的并行使用，会破坏骨架的连接性，因此又增加了 6 个 4×3 的保留模板。图 4 是其中三个，另外三个由图中的模板逆时针旋转 90° 得到。

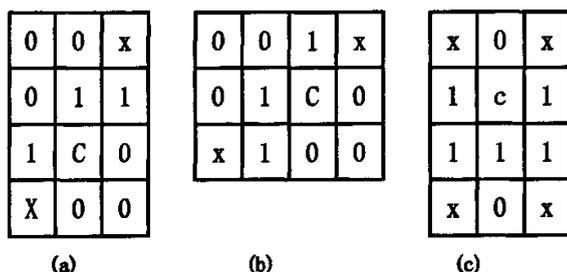


图 3-8 处理后的保留模板

Fig 3-8 Mask after processing

由于所有 4 个方向的细化模板和保留模板是同时并行作用于目标，所以产生的骨架是各向同性的，即若目标旋转 90° 的整数倍时，细化产生的骨架是相同的。图 3-9 为二值化和细化后的图像。



图像 3-9 二值化和细化后的图像

Fig 3-9 Image after binarize and thinning

3.2.6 指纹节点的提取

本课题中节点提取采用的模板是见公式 3.14:

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 10 & 1 \\ 1 & 1 & 1 \end{pmatrix} / 10 \quad (3.14)$$

下面为节点提取具体算法实现:

```
//首先 读取指纹图像转化成的矩阵
rows = image->Vres;
cols = image->Hres;
img2->Hres = image->Hres;
img2->Vres = image->Vres;
img2->Size = image->Size;
img2->i = new Pixel[image->Size];
img2->p = new Pixel*[image->Size];
for(int x=0;x<image->Hres;x++)
img2->p[x] = new Pixel[image->Vres];
```

```
//模板采用公式 (3.11)
float mask[3][3] = {{0.1, 0.1, 0.1}, {0.1, 1, 0.1}, {0.1, 0.1, 0.1}};
float value;
char mesg[256];
int i, j, k, l;
for(i=1; i<rows-1; i++)
{
    for(j=1; j<cols-1; j++)
    {
        value=0;
        for(k=-1; k<2; k++)
        {
            for(l=-1; l<2; l++)
            {
                value = value +
(float)((int)(image->p[i+k][j+l])*mask[k+1][l+1]);
            }
        }
    }
}
// 如果 mag 是 1.1 说明只有一个临近的节点存在 那就表明这个节点是端点
if((int)(value*10) == 11)
{
    sprintf(mesg, "ridge end \n");
    img2->p[i][j] = 1;
}
// 如果是1.3 说明 2 两个临近节点存在 那即表明此点为分叉点
if((int)(value*10) == 13)
{
    sprintf(mesg, "ridge bifurcation \n");
```

```
img2->p[i][j] = 1;
}
}
}
for(int r=0;r<img2->Vres;r++)
{
    for(int s=0;s<img2->Hres;s++)
    {
        image->p[r][s] = img2->p[r][s];
    }
}
```

图 3-10 为提取节点后的图像:



图 3-10 点提取后的图像

Fig 3-10 image after minutiae extraction

3.3 指纹的匹配

指纹是比较复杂的, 指纹识别算法最终都归结为在指纹图像上找到并比对指纹的特征点。对细化后的指纹图像提取特征点数据, 并去除伪特征点, 将处理结果与指纹数据库中已有的模板数据进行匹配识别。指纹的匹配问题一直是自动指纹识别系统研究中的核心问题。过去三十年内, 人们在这个领域进行了大量的研究, 最常见的是基于细节点特征的指纹匹配算法和基于指纹纹理特征的匹配, 但是仅仅使用纹理特征进行匹配的算法识别率都是比较差的, 远不如使用细节匹配算法的效果。

本课题也是采用基于细节节点的匹配。

本课题采用了基于四分叉的指纹匹配算法,单纯使用指纹细节特征进行匹配存在两个问题:首先指纹细节特征的提取受指纹图像的质量影响很大,这会导致指纹识别系统的拒识率明显上升。其次只使用指纹细节特征无法处理指纹的形变问题。很多学者提出这样一种观点即采用混合细节节点匹配和纹理特征匹配的算法,取得了较好的匹配效果,这也是下一步研究改进的方向。

指纹识别算法的具体实现代码:

```
SQuad *q1 = NULL, *q2= NULL;
//首先把指纹节点特征转化成四叉树
q1 = makeQuadTree( minArr1, minCount1 , 0, 0, 255, 255);
q2 = makeQuadTree( minArr2, minCount2 , 0, 0, 255, 255);
//返回四叉树比较的结果
return compareQuadTrees( q1, q2 , 0) / (compareQuadTrees( q1, q2 , 0) +
compareQuadTrees( q2, q2 , 0) ) * 200.0f;
//四叉树比较的具体实现
int i;
float difference = 0.0f, returnVal = 0.0f;

if( q1 == NULL || q2 == NULL )
    return 13.0f;

for( i=0; i<4; i++){
    //difference += (abs( q1->score[i] - q2->score[i] )) * (level + 1) ;
    difference += pow( (float)(abs( q1->score[i] - q2->score[i] )) , (level
+ 1) );
}

returnVal = 1/(difference * pow(4.000, level) + 1);
```

```
for(i=0; i<4; i++ )
{
    returnVal += compareQuadTrees( q1->quad[i], q2->quad[i], level +1 );
}
return returnVal;
```

3.4 指纹数据数据库

3.4.1 为何采用 Xml 建立数据库

3.4.1.1 什么是 xml

XML(eXtended Markup Language)是从 SGML(Standard Generalized Markup Language)进化来的,最初 SGML 是为了解决文档及其格式问题的一种标记语言,所以非常的复杂和难以使用,那时计算机的应用水平还处在很低的阶段,并且那时人们只需要传递和显示一些简单的数据。因此,89年 Tim Berners-lee 依据 SGML 开发出一种超文本格式,就叫 HTML(HyperText Markup Language),非常的简单。当计算机的应用水平也大幅度的提高的时候,人们已经对太简单的 HTML 开始不满意,因此又开始对其进行修改、升级,就这样 HTML 从 1.0 一直升到 4.0 的版本,扩展了许多的命令,实际上也是为了让 HTML 解决更多的问题,但是因为 HTML 本身的缺陷,为了解决更多的问题,是不是会把简单的 HTML 变成一个非常复杂的,难以理解的语言,就连现在看一个 HTML 的源文件,都会头大好几圈,再加上 HTML 是把数据和显示格式一起存放的,如果我们只想使用数据而不需要格式,可以想象,分离这些数据和格式是多么的困难。而随着 Internet 的发展,98年1月 W3C 公布了 XML1.0 版本,立即成为计算机史上一个重要的里程碑。XML 的文件是以树状方式存储,同时也有属性,这非常符合面向对象方面的编程,而且也体现了以对象方式存储,ORACLE 数据库就使用了这种面向对象的特性,但是 ORACLE 数据库的根本是关系型的数据库,因此速度和稳定性都有一定的问题,而且特别的复杂,所以大家基本上很少使用它的这一特性,深怕跳到坑里去! XML 可扩展标记语言已是各种数据特别是文档的首选格式,国际主流的数据库厂商们自然也随行就市,全都推出了兼容传统关系型数据与层次型数据(XML 数据)混合应用的新一代数据库产品。2006 年的这一特点更加明显。IBM 公

司新推出的 DB2 9 版本中，直接把 XML 作为其新产品的最大卖点；微软和 Oracle 也同时宣传它们的产品也可以实现高性能 XML 存储与查询，使现有应用更好的与 XML 并存。2006 年市场的需求和厂商的举动意味着 XML 将是未来数据库发展的主体趋势。

3.4.1.2 XML 与数据库

诞生于二十世纪 60 年代的数据库技术及其应用系统，在经历了层次数据库、网状数据库到关系数据库、面向对象数据库的发展后，已成为计算机信息系统中的重要基础和支柱。但是，上个世纪 90 年代以来，Internet 的飞速发展使人类社会的信息量迅速膨胀，对信息管理和信息交换的需求更是空前强烈和迫切，使得已在传统的商业处理领域内趋于成熟的数据库技术，面临着许多新的问题和挑战。与此同时，XML 技术诞生并迅速成长起来，在诸多领域得到广泛的支持，有着广阔的应用前景，在一定程度上形成了对数据库技术的挑战，目前有不少将 XML 用于数据库应用的研究正在进行中。

数据库技术面临的挑战

传统数据库的应用领域主要是针对商业与事务处理。Internet 时代到来以后，Web 改变了人们习惯的处理方式，也给数据库技术提出了一个必须面对的重要问题：即如何有效地存储和管理 Web 上以指数增长着的数据（文档），使它们既能被高效地操作和维护，又能在 Web 这个平台上方便地表示和交换。

当前，典型的问题有：

1. 将现有数据库中数据转换成适于 Web 的形式，需要相当的工作量。时至今日，虽然从 CGI、Server API、Scripts 到 Java 等等出现了许多解决方法，并且已在实际应用中取得了很好的效果，但仍有问题存在。

2. 传统数据库中的数据以二进制码的形式存储，并且不同的数据库管理系统有各自的专有格式，这既带来了表示上的困难，更是进行交换的一大障碍。而随着网络的发展，数据交换的能力已成为新的应用系统的一个至关重要的指标。

当然，我们也已经看到，数据库技术正在积极地吸收和利用 Web 方面的新技术。当前主流的数据库产品都宣布了对 XML 的支持。但是，数据库管理系统（DBMS）与 XML 的这种捆绑是否能够充分发挥效率，还需要在实际中检验。

3.4.1.3 XML 与数据库技术的比较

从技术角度讲, XML 和关系数据同属于数据管理的手段。狭义的 XML 仅仅指一种语言和采用该语言所描述的 XML 文档, 广义的 XML 包括 XML 语言、XML 文档以及所有与 XML 相关的工具和技术, 例如, XML 解析器及解析技术, XML 转换技术 (XSL、XSML) 等。广义的 XML 与 DBMS 大致具有相似的作用, XML 与 DBMS 相同之处在于:

1. 提供数据存储。关系数据以数据库系统为手段, XML 以文件系统为手段。

2. 提供对数据的直接存取访问。两者都不需要用户关心数据的物理结构。

3. 提供数据的模式描述。XML 采用 DTD (Document Type Definition, 文档类型定义) 或 XML Schema (XML 规范) 来描述数据的逻辑结构; 关系数据通过关系模式来描述数据的逻辑结构。

4. 提供应用逻辑接口。XML 采用 SAX (Simple API for XML) 和 DOM (Document Object Model) 定义应用编程接口, 使应用程序能够访问和更新 XML 文档的样式、结构和内容; 关系数据采用 ODBC、JDBC、OLE DB 等。

作为两种相近的技术, 两者也存在很大差别:

相对 XML, 关系数据库的优势在于: 技术成熟、应用广泛; 数据管理能力强 (包括存储、检索、修改等); 数据安全程度高; 稳定可靠的并发访问机制等。另一方面, 随着 Web 技术的不断发展, 信息共享和数据交换的范围不断扩大, 传统的关系数据库也面临着挑战。第一, 数据库技术的应用是建立在数据库管理系统基础上的, 各数据库管理系统之间的异构性及其所依赖操作系统的异构性, 严重限制了信息共享和数据交换范围。第二, 数据库技术的语义描述能力差, 大多通过技术文档表示, 很难实现数据语义的持久性和传递性, 而数据交换和信息共享都是基于语义进行的, 这样在异构应用数据交换时, 不利于计算机基于语义自动进行正确数据的检索与应用。第三, 数据库属于高端应用, 需要昂贵的价格和运行环境等。

归结起来, 数据库在数据管理方面具有管理方便、存贮占用空间小、检索速度快、修改效率高、安全性好等优点, 但客观上需要用一种应用方式将其丰富的数据有效地发布出来, 以消除平台差异、增强语义描述功能、降低环境要求。

相对于数据库技术，XML 技术在数据应用方面具有很多优点。第一，跨平台。XML 文件为纯文本文件，不受操作系统、软件平台的限制；第二，易表义。XML 具有基于 Schema 自描述语义的功能，容易描述数据的语义，这种描述能为计算机理解和自动处理。第三，XML 不仅可以描述结构化数据，还可以有效描述半结构化，甚至非结构化数据。但另一方面，XML 技术在数据管理方面存在明显缺点。首先，XML 技术采用的是基于文件的管理机制，文件管理存在着容量大、管理困难的缺点；第二，目前 XML 的检索是基于节点的检索，存放大量甚至海量数据的 XML 文件造成检索速度极低；第三，解析手段有缺陷。XML 具有两种解析机制，SAX 方式是基于文件的解析，速度慢，DOM 方式是基于内存的方式，资源消耗极大；第四，修改效率低，目前 XML 的修改是基于节点的；最后，XML 的安全性及并发操作机制也是需要解决的问题之一。表 3-1 为各种数据存储技术的比较。

表 3-1 各种数据存储技术比较

Table 3-1 Comparisons of Database

存储方法	存储格式	主要优点	主要缺点
文件系统	ASCII 文件	易管理、易存储	访问和更新困难
关系数据库	表	可扩展、可靠性、安全性	简单的查询也需要重链接来实现
OO/OR 数据库	表和对象	易实现、支持抽象数据类型	Xml 分解困难，无法映射没有 DTD 或 Schema 的 xml 文件
Native XML 数据库	特殊数据类型	具有灵活性，高速度	尚不成熟

总体上看，XML 在数据应用方面上具有易表义、跨平台等优势，但客观上需要一种有效的存储、检索和修改等方面的管理机制。

3.4.1.4 发展趋势

通过上述分析比较不难得出，在数据管理方面数据库尤其是关系数据库具有其他方式（包括普通文件、XML 文件、对象数据库）无法比拟的优越特性。而在应用方面，尤其是在 Web 信息共享及异构应用数据交换方面，XML 又具有其他技术无法比拟的优点。通过

比较,有这样几点结论:

1. 人们在现实中还是习惯用关系数据库管理数据,绝非仅仅习惯使然,更深层的原因是 XML 缺乏像关系数据库这样有效的数据管理手段,造成数据存贮管理困难、检索修改效率极低,这同时也是制约 XML 技术普及的因素之一,XML 尚不能代替关系数据库。

2. 通过 XML 技术降低数据库的使用条件、可以使其中丰富的数据资源得到利用,并且 XML 在异构数据交换方面具有很大优势。

3. XML 和关系数据库在数据应用和数据管理方面各有优势,应是一种相辅相成、扬长避短的关系。

从技术角度讲,“以数据库为存贮手段,以 XML 为交换载体”的数据管理模式是一种趋势。XML 存放在关系数据库中,XML 数据以虚拟 XML 文档的形式存在。从用户的角度,XML 文档和数据库之间的区别是不透明的。数据库中存在的数据库虚拟的 XML 文档,对用户而言,重要的不是数据存在何处,而是数据的逻辑结构合理性和操作界面的方便性。

目前,虽然处理和操作 XML 文档的开销要显著地大于关系数据库系统中类似的表。其中,对 XML 文档进行装载和转储的开销占了很大的比例。XML 文档的长度对其处理开销有非常明显的影响。但无论如何,XML 正在迅速地发展和不断地完善中,前景非常光明。例如,正在开发和标准化之中的 XML 查询语言(XQL)即是针对 XML 在数据库方面的应用。同时,在这个领域内,还有很多问题有待我们去研究和解决。

3.4.2 本软件数据库的设计

3.4.2.1 CMarkup 介绍

本课题采用开放源代码的 Cmarkup xml 解析器来完成 xml 指纹数据的储存,索引,查找,删除功能。Cmarkup 是基于“压缩”文档对象模型(EDOM:“Encapsulated Document Object Model”),这是这个简单 XML 处理器的关键。它的一套 XML 处理方法和产生的效果与 DOM(Document Object Model)是一样的。但是整个 DOM 有无数的对象类型,而 EDOM 只定义了一个对象:XML 文档,EDOM 回到了 XML 原始的吸引力—简单。CmarkUp 封装了 XML 文档文本、结构和当前位置。它提供了增加元素、导航和得到元素属性和数

据的所有方法以。在文档中操作被执行的地方取决于当前位置和当前孩子位置。当前位置允许你增加一个元素到文档中的这个位置，而不需要明确指出增加对象到文档的这点上，在 CMarkup 的有效生命周期里，对象一直保持着—一个用来描述文档的字符串，当你调用 GetDoc。可以得到它。

3.4.2.2 本软件数据库设计

本软件数据库是基于 ASCII 文件存储的，指纹数据的新建、更新、查找和删除都利用 CMarkup 本身提供的功能。

3.5 软件性能分析

指纹识别系统的性能评价体系应该从传感器、识别算法和应用系统的各个方面去研究。对传感器来说，考察其质量的因素包括噪声水平、畸变水平、图像分辨率、图像清晰度（对比度）、体积、安装调试方便性及产品的稳定性、一致性、成本等；对应用系统来说，作为一个精密的电子产品，应该有友好的人机界面，使用维护方便，功能齐全，安全可靠，并且便于产品更新换代。

指纹识别系统的性能评价的最重要的方面是指纹识别算法的性能评价，笔者认为，评价算法性能的基础首先是统一的指纹数据库的建立，通用的指纹数据库应该尽可能满足以下条件：

1) 数据库应该有一定的人群覆盖面，人群的选择应该兼顾职业、年龄及性别的变化，因为这些因素对指纹的质量分布有较大影响。

2) 数据库应该包含同一枚指头不同次采样的指纹，不同次采样应该在一定的时间间隔后再采样（比如几天、几周甚至几个月、几年，越远越好）。

3) 数据库应该有足够大的规模。比如 100 个指头，每个指头采样 10 次，共 1000 幅指纹图像，在建立了指纹数据库后，每一幅指纹图像都需要进行特征提取，以建立合法用户的档案。指纹性能的测试可以采用每一幅指纹特征与数据库中其他所有的指纹特征进行比对的方法进行，如果识别算法是“对称的”，即“指纹 A 与指纹 B”的比对结果与“指纹 B 与指纹 A”的比对结果相同，则整个测试不需要循环进行，只需按照数据库的自然顺序逐个进行即可。以上述“100 个指头 10 个采样 1000 个指纹”的数据库为例，

共需比对次数为 50 万次。在理想状态下，其中应该有 450 次正确的比对通过。

指纹算法的具体性能评价指标有：

1) 拒真率 (FRR) 即真正用户在建立档案后无法通过的概率，可以用同一指头的多次采样指纹，被自己的建档拒绝的概率来测量。拒真率越低越好。

2) 认假率 (FAR) 即非法用户能通过指纹识别系统的概率，可以用多次比对中，发生两个不同指头的采样却通过识别检测的概率来测量，认假率越低越好。

3) 拒登率 (FER) 即在普通人群中，一枚指头由于采集质量、磨损程度、特征点多少等任何原因不能够建立档案进行后续识别的概率，对这类指纹进行勉强的自动识别，将得不到起码的 FRR 和 FAR 性能要求，拒登率越低越好。

4) 平衡点即调整指纹识别的门限，使拒真率和认假率相同，平衡点越低越好。

5) 建档时间 即平均每一枚指纹进行指纹图像采集、图像处理、特征提取和特征建库的时间，建档时间越短越好。

6) 比对效率 即在建档后，直接比对一对指纹特征（不进行图像处理和特征提取）所用的时间，一般用每秒钟比对的次数来表示，当然越多越好。

7) 特征文件字节数即平均每一枚建档指纹所用的特征文件长短，用字节表示，越短越好。

8) 法通用性即指纹识别算法对指纹传感器的敏感性，需要考虑当传感器图像尺寸、噪声水平、失真程度、手指摺印角度等因素改变时，识别算法的适应性或程序修改的工作量，这一条暂时没有具体衡量指标。

其他软件工程指标包括程序可靠性、易用性、界面友好、统计容易看出，上述指标互相是有矛盾和抵触的，单一指标衡量算法性能很可能是片面的、不科学的，所以，从全面的角度衡量性能指标是有必要的。

3.5.1 性能分析体系介绍

目前，指纹识别技术的研究无论是在前端的数据采集上还是在后端的指纹识别算法上都已经取得了巨大的进展。于是，很多人认为现在的指纹识别技术已经很完善了，不再需要研究了。这种观点是错误的，在指纹识别技术上仍然有一些没有解决好的问题，如：

低质量指纹图像的处理、形变指纹图像的匹配、活体指纹的检测等。这些都是经常遇到、非常重要和亟待解决的问题。指纹识别系统系统的性能评价体系应该从传感器、识别算法和应用系统等多个方面去研究。由于本课题只是指纹软件的开发，指纹识别算法性能评价就成了评价指纹识别软件的重要指标。评价算法性能的基础首先是要建立一个具备各种多样性的指纹数据库，建立的指纹数据库应该尽可能有一定的人群覆盖面，人群的选择应该兼顾职业、年龄及性别的变化，因为这些因素对指纹的质量分布有较大影响。同时数据库应该包含同一枚指头不同次采样的指纹，不同次采样应该在一定的时间间隔后再采样（比如几天、几周甚至几个月、几年，越远越好）。最后是数据库应该有足够大的规模，比如，100个指头，每个指头采样10次，共1000幅指纹图像。

在建立了指纹数据库后，每一幅指纹图像都需要进行特征提取，以建立合法用户的档案，指纹性能的测试可以采用每一幅指纹特征与数据库中其他所有的指纹特征进行比对的方法进行。目前在国际上有两个比较权威的指纹认证和评测方法：指纹识别竞赛 FVC^[27,28,29,37]和商用指纹算法评测FpVTE^[38]，FVC是由意大利博洛尼亚大学、圣何塞州立大学以及密歇根州立大学等组织的指纹识别算法的国际竞赛，受到了来自世界各国企业和研究机构的普遍关注，并得到了国际指纹业界的认可和接受，具有非常广泛的影响力，也是在指纹识别算法方面唯一一个世界性大赛。商用指纹算法评测FpVTE2003是由美国国家标准局NIST所组织的指纹识别评测。目的是为了评测指纹识别算法的匹配、验证、识别的准确性。FpVTE2003将会作为入境美国进行指纹验证时的法定指纹评估标准。FpVTE的结果将会作为大规模指纹识别技术的基础。

3.5.1.1 FVC2004 的指纹评估标准

FVC从2000年开始，每两年举办一次，大会提出的指纹评估标准每届也都会有相应的变化，随着每年参加组织的个数逐年递增，已经成为评价世界各个国家指纹识别算法发展水平的国际权威性评价机构。最近的一届是FVC2006，由于我们主要是介绍指纹评估标准，而FVC2006还并没有公开大会的评估标准，因此下面我们主要介绍FVC2004也就是第三届大会的指纹评估标准。

FVC2004提供了open和light两种竞赛方式，其中light方式在注册和匹配时间(模板大小以及内存占有方面有比较严格的限制，而open是开放式，有更广泛的适应性，在

open 类测试中匹配时间要小于 5 秒, 算法的计算资源受到了限制, 限制条件如下: 注册时间<0.5s; 匹配时间<0.3s; 模板尺寸<2kb; 分配的内存量<4kb; 算法测试的数据库包括 4 个不同的指纹库, 其中包括用 3 种不同的指纹采集仪所建立的指纹库和一个用算法生成的指纹库。每个库的指纹尺寸都不相同, 包括 110 个手指, 每个手指 8 幅图像。对每个指纹库, 真匹配 (来自同一个手指间的图像匹配的数目是 2800 次) 而假匹配 (不同手指图像间的匹配%的数目是 4950 次) 对参加竞赛的每个算法测试的性能指标^[27]包括:

- (1)注册拒绝率, 匹配拒绝率
- (2)真、假匹配的分直方图等
- (3)错误率
- (4)平均匹配时间和平均注册时间
- (5)为增强和匹配分配的最大内存
- (6)平均及最大模板尺寸

3.5.1.2 FpVTE2003 的指纹评估标准

和FVC相比FpVTE2003^[28]有很多特色,它反映了算法的其他方面性能FpVTE包括3个不同规模的测试:大规模测试、中规模测试、小规模测试^[28]。表3-2阐明了每个测试的相关环境。

表3-2 每个测试的相关环境

Table 3-2 Every environments of counterpart tests

	LST	MST	SST
手指模式	多个手指	单个手指	单个手指
指纹模式	滚动的、平面、掌印	掌印和平面	平面
样本模式	64000套第1~10个指纹	10000枚指纹(都是右手食指)	1000枚指纹(都是右手食指)
时间限制	21	14	14

性能通过下列指标评测: (1) 匹配和不匹配分布; (2) Roc(t);(3)切片图; (4)指纹质量影响; (5)手指模式的影响等等。

FVC在作指纹识别算法性能测试时，并无外界指纹输入，是使用标准的指纹图像库来测试的。

3.5.1.3 本课题的指纹评估标准

本课题指纹识别算法评估标准主要参考FVC2004中。评价指标为FER(拒登率),平衡点,建档时间, 比对效率, 特征文件字节数。其中平衡点即调整指纹识别的门限, 使拒真率和认假率相同, 平衡点越低越好。数据库采用FVC2004官方的DB1^[7]数据库。在DB1数据库中, 有些指纹图像的形变就很大, 而位于第一名的指纹识别算法的等错误率(EER)是1.97%, 从中可以看出形变指纹图像的处理仍然不是很理想, 有着很多工作要做。

3.5.2 性能分析结果

表3-3为本软件算法性能测试结果, 可以看出基本了常用的指纹识别的需要, 与官方获奖算法还有很远的距离, 很多方面的性能还需要提高。

表3-3 DB1数据库的本软件算法性能测试

Table 3-3 Software test Results using DB1 database

FER(拒登率)	平衡点	建档时间	比对效率	特征文件字节数
5.89%	6.77%	平均 0.36s/枚	平均 83.5 次/秒	平均 65.3kb/枚

本章在前一章的基础上, 实现了高性能的图像增强算法, 并把提取后的节点存入下存入数据库, 最后应用几种评价指标评价了软件。取得了满意的结果。

全文结论

本论文以指纹识别中指纹图像增强算法，节点提取与匹配算法为理论基础，进行了自动指纹识别软件的研究与开发。其主要的研究内容和相关的结论如下：

1、对低质量指纹图像的传统指纹图像增强算法进行了改进。传统的指纹图像增强算法主要依据沿脊线垂直方向的灰度变化呈正弦波形的假设，设计具有方向选择性的滤波器。因为沿着垂直于纹线的方向来看，指纹纹线大致形成一个二维的正弦波，除了模式区等少数异常区域以外，在一个小的局部区域内，指纹纹线的分布具有良好的频率特性和方向特性。Gabor函数具有最佳时域和频域连接分辨率的特点，是唯一能够达到时频测不准关系下界的函数，可以很好地兼顾时域和频域信息，但是由于Gabor滤波计算量大、运算复杂度高，直接使用很难满足实际工程的需要。为此本文对Gabor滤波实现方式进行了改进：

1)考虑到指纹的方向和脊频都具有局部一致性，就没必要对图像中的每一点进行统计。改进的做法是将图像分割成大小为的小块，以该小块中心点的方向和脊频作为该小块中每一点的方向和脊频。

2)对Gabor滤波器的复杂度最大的两个参数(方向和脊频)进行量化，这样就可以用有限个模板满足整个滤波的要求。

2、实现了移植性良好的指纹识别软件。

目前采用指纹数据库的系统中都是采用成熟的商用数据库管理系统，这样就给指纹识别系统的部署带来了很大的难题。建立了基于xml文件的指纹数据存储文件，可方便的移植到现有成熟的各大主流商用数据库管理软件中去，同时也为识别大规模基于网络认证的指纹安全系统提供了可能。

进一步研究的方向：

由于时间和人力所限，对指纹识别只做了以部分的工作，还需要以后对以下问题进行进一步探讨：

1、在软件中指纹识别模块实现高性能采用遗传算法的指纹分类算法；

目前只是实现了指纹识别中1:1的指纹匹配功能，没有实现指纹识别系统中1: N的识别功能，在下一步的工作当中，要实现基于指纹特征的高性能指纹分类算法。

2、指纹图像数据库数据的加密实现；

指纹特征存储只是基于简单的xml文件，在安全性等各方面的指标还不能与商用的数据库管理系统比较，指纹特征数据完全暴露在用户面前，下一步应该完善用户访问控制方面的功能，同时下一步应该继续关注Native XML等基于xml数据库技术的发展与时俱进改进本课题的数据存储技术。

3、指纹匹配算法还有提升的空间

指纹匹配算法当中仅仅实现了最基础的指纹匹配算法，下一步还有要提升的空间。

4、具体应用上的研究；

目前仅仅只限了理论的开发，距离实际应用，还有一段距离，下一步应该在本课题研究成功的软件的基础下继续探讨在各个行业里的实际应用。

参 考 文 献

- [1] 戴平阳. 指纹识别技术研究进展[J]. 厦门大学学报(自然科学版), 2002(6): 750-755.
- [2] 张海春, 回文博, 林立忠. 指纹识别技术研究进展. 石家庄学院学报, 2005, 5, (3): 25-28.
- [3] 田捷 陈新建 张阳阳 杨鑫等. 指纹识别技术的新进展[J]. 自然科学进展, 2006, 16(4): 400-406 页.
- [4] Lawrence O' Gorman. An overview of fingerprint verification technologies [J]. Information Security Technical Report, 1998, 3(1): 21-32
- [5] 马笑箫. 指纹自动识别系统中的关键技术一方向图[J], 重庆大学学报(自然科学版), 2001, 24(3): 91-94
- [6] 罗希平 田捷. 自动指纹识别中的图像增强和细节匹配算法. 软件学报, 2002(5): 946-956.
- [7] 洪华军 乔为民 朱立新. 指纹图像自动识别系统预处理算法的研究[Z], 兰州中科院近代物理所
- [8] Chikkerur S, Wu C H, Govindaraju V. A systematic approach for feature extraction in fingerprint images[R]. In: ICBA 2004. Florida, USA, Dec 16-19, 2004: 344~350.
- [9] Lin H, Wan Y, Jain A K. Fingerprint image enhancement: Algorithm and performance evaluation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1998, 20(8): 777-789
- [10] Areekul V, Wachareeruetai U, Tantaratana S. Fast separable gabor filter for fingerprint enhancement[R], In ICBA 2004. Hong Kong, China, July 15-17, 2004: 403-409
- [11] 刘旭, 田捷 自动指纹识别算法在嵌入式系统上的实现[Z]
中国科学院自动化研究所复杂系统与智能科学实验室, 北京 100080
- [12] Sherlock B, Monro D, Millard K. Fingerprint enhancement by directional Fourier filtering[J]. IEEE Proceedings-Vision, Image and Signal Processing, 1994,

参考文献

- 141(2):87-89
- [13] He Y L, Tian J, Luo X P, et al. Image enhance and minutiae matching in Fingerprint Verification[J]. Pattern Verification. Pattern Recognition Letter, 2003, 24: 1349-1360
- [14] Ceng G Z, Hall R W. Parallel thinning with two-subiterational algorithms[J]. Communications of the ACM, 1989, 32:359-373
- [15] Chikkerur S, Wu C, Govindaraju V. A systematic approach for feature extraction in fingerprint images[R]. In: ICBA 2004. Hong Kong, China, Dec17-19, 2004:344-350
- [16] Luo X P, Tian J. Knowledge based fingerprint image enhancement. In: ICPR 2000. Spain, Barcelona, Sep 3-7, 2000:783-786
- [17] Luo X P, Tian J, Wu Y. A minutiae algorithm in fingerprint verification[R]. In: The 15th ICPR. Barcelona, Spain, Sep 3-8, 2000:833-836
- [18] 冈萨雷斯. 数字图像处理(第二版) [M]. 阮秋琦, 阮宇智, 等译. 北京: 电子工业出版社, 2004.
- [19] Lee H C, Gaensslen R E. Advances in Fingerprint Technology, 2nd ed, New York: Elsevier, 1991
- [20] Q XIAO, H RAAFAT. Fingerprint image postprocessing: a combined statistical and structural approach[J]. Pattern Recognition, 1991, 24(10):985-922
- [21] L HONG, Y WAN, A K JAIN. Fingerprint image enhancement: algorithms and performance evaluation[J]. IEEE Trans. Pattern Analysis Machine Intelligent, 1998, 20(8):787-789
- [22] 解梅, 马争. 基于脊向指纹滤波算法. 电子学报, 2004, 32(1): 5-7.
- [23] 王杰文, 李赫男. XML 数据库技术研究进展. 南华大学学报(自然科学版), 20(3):42-46
- [24] 卢佳等. 图像数据库索引技术的研究与探讨. 计算机工程与设, 2005, 26(7): 1759-1761
- [25] 周长发. 精通 vc++图像处理编程[M], 电子工业出版社 2002 年 4 月

- [26] 卢朝阳, 张岗山, 刘琳. 指纹识别系统性能评价方法. 西安电子科技大学学报(自然科学版), 2002, 29(6):804~808.
- [27] 漆远, 田捷, 邓翔. 基于遗传算法的指纹图匹配算法[J]. 及应用. 软件学报, 2000, 4:488-493.
- [28] Bazen A M, Gerez S H. Fingerprint matching by thin-plate spline modeling of elastic deformations[J]. Pattern Recognition, 2003, 36(8):1859~1867
- [29] Biometric system lab, Pattern Recognition and Image processing Laboratory[Z], Biometric Test Center, [Http://bias.csr.unibo.it/fvc2000/](http://bias.csr.unibo.it/fvc2000/)
- [30] Biometric system lab, Pattern Recognition and Image processing Laboratory [Z], Biometric Test Center, [Http://bias.csr.unibo.it/fvc2002/](http://bias.csr.unibo.it/fvc2002/)
- [31] FpVTE2003[Z], <http://fpvte.nsit.gov>
- [32] [Http://bias.csr.unibo.it/fvc2004/categories.asp](http://bias.csr.unibo.it/fvc2004/categories.asp)
- [33] 卢朝阳, 张岗山, 刘琳. 指纹识别系统性能评价方法[J], 西安电子科技大学学报(自然科学版), 2002, 29(6):74-78.
- [34] 程建刚 田捷 何余良. 基于非线性扩散滤波的指纹增强算法, 生物特征识别研究组, 复杂系统与智能科学重点实验室, 中科院自动化所, 100080, 北京
- [35] 罗希平 田捷. 自动指纹识别中的图像增强和细节匹配算法[Z], 北京 中国科学院自动化研究所人工智能实验室 100080
- [36] 任群 张晓鹏 田捷. 指纹图像质量自动评测方法[Z], 中国科学院自动化研究所复杂系统与智能科学实验室, 北京 100080
- [37] Biometric system lab. Pattern Recognition and Image processing Laboratory [Z], Biometric Test Center, [Http://bias.csr.unibo.it/fvc2004/](http://bias.csr.unibo.it/fvc2004/)
- [38] Federal Bureau of Investigation. The Science of Fingerprints: Classification and Uses[Z]. Washington, D C, USA: GPO, 1984

攻读硕士期间发表论文

[1]杜宝雷, 曾建谋等. 指纹识别系统规模化应用的研究. 机电工程技术. 2006年第12期: 86, 87, 102.

致 谢

本论文是在导师曾建谋教授的细心指导和热情关怀下完成，在硕士学习与科研期间，他渊博的知识、严谨的治学态度、诲人不倦的师德和实事求是的治学精神令我受益匪浅，在生活中对我悉心照顾，并给我无微不至的关怀。在成文之际，谨向曾老师致以最衷心的感谢和最诚挚的敬意，祝愿他身体健康，合家幸福！

感谢我的师弟叶国林和各位同窗，在课题组以及学习期间相互探讨，相互学习，相互促进，为我的科研课题增添活力，注入新的思维。

最后，我要深深地感谢远在千里之外的父母以及其他亲人和朋友，他们多年来含辛茹苦，一如既往地给予我物质和精神上的巨大支持，是你们的支持和鼓励给了我无比的信心和勇气。