

硕士学位论文

基于汗孔的高分辨率指纹匹配算法

HIGH RESOLUTION FINGERPRINT MATCHING BASED ON PORES

方博

哈尔滨工业大学

2011 年 12 月

国内图书分类号:TP391.4

国际图书分类号: 004.9

学校代码:10213

密级: 公开

工学硕士学位论文

基于汗孔的高分辨率指纹匹配算法

硕 士 研 究 生: 方博

导 师: 徐勇教授

申 请 学 位: 工学硕士

学 科: 计算机科学与技术

所 在 单 位: 深圳研究生院

答 辩 日 期: 2011 年 12 月

授予学位单位: 哈尔滨工业大学

Classified Index: TP391.4

U.D.C: 004.9

Thesis for the Master Degree in Engineering

HIGH RESOLUTION FINGERPRINT MATCHING BASED ON PORES

Candidate:	Fang Bo
Supervisor:	Prof. Xu Yong
Academic Degree Applied for:	Master of Engineering
Speciality:	Computer Science and Technology
Affiliation:	Shenzhen Graduate School
Date of Defence:	December, 2011
Degree-Conferring-Institution:	Harbin Institute of Technology

摘 要

近年来,光电技术突飞猛进,指纹采集仪器的功能也越来越强,高分辨率指纹识别逐渐走入人们的视野。相比传统的指纹识别,高分辨率指纹识别不仅能够提取更高分辨率的指纹细节信息,还具有高准确率,防造假等优点。但是高分辨率指纹识别面临复杂度较高,受非线性形变影响较大等问题。虽然在高分辨率下可获得更精细的指纹特征,但是指纹的局部特征在被放大的同时,指纹局部形变也被放大,这对汗孔这类第三层指纹特征的匹配带来较大影响。目前的第三层指纹特征匹配算法大多采用整体匹配,没有考虑到局部形变造成的误差。

本文针对目前高分辨率指纹匹配的问题,提出了两种基于分治思想的汗孔匹配算法。

第一种算法为基于细节点三角剖分分块的汗孔匹配算法,算法首先根据细节点进行匹配,然后基于匹配细节点进行三角剖分,对指纹进行分块。然后分块匹配每个块里的汗孔。分块匹配使复杂度大大降低,并一定程度上缩小了非线性形变对匹配结果的影响。在得到汗孔的匹配结果后,本文又结合细节点,方向场信息进行了匹配融合,匹配结果相比同类算法有一定提高。

第二种算法为基于脊线的汗孔匹配算法。首先根据脊线进行匹配,通过计算相似度找到相似的脊线,然后根据距离计算匹配的脊线,并计算脊线的匹配得分。然后根据匹配的脊线,找到脊线上的汗孔,并采用动态规划法计算匹配得分,最后的得分为脊线得分与相应汗孔的匹配得分的积。然后结合方向场和细节点进行匹配级融合。采用基于脊线的汗孔匹配可以有效解决局部脊线形变带来的误差,一定程度上提高了指纹匹配的准确率。

最后在香港理工大学的高分辨率指纹库上进行了实验。实验表明本文提出的两种高分辨率指纹识别匹配算法相比目前已提出的同类算法性能有所提升。

关键词: 高分辨率指纹识别; 汗孔匹配; 分治; 融合

Abstract

In recent years, with the development of photovoltaic technology, high resolution fingerprint gradually came into people's vision. Compared to traditional fingerprint identification, high resolution fingerprint recognition is able to extract more features, which can prevent fraud and improve the accuracy. The major problems of high resolution fingerprint identification are the high complexity, influence of the nonlinear deformation, and so on. Since the size of high resolution fingerprint is very big, the local features are magnified. Therefore the deformation is magnified, which will affect the accuracy of high resolution fingerprint identification. At present, most of the high resolution fingerprint algorithms do the matching using whole features, which will be affected by the local deformation. That will decrease the accuracy of fingerprint matching.

In order to solve the problem of high complexity and local deformation, two algorithms are proposed based on the thought of DC (divide and conquer). One method we proposed is to match the pores block by block. In the first step, fingerprint is divided into blocks using triangulation based on matched minutiae. Then the pores in each block are matched based on distance. This method can greatly reduce the complexity, and the effect of the local deformation to the pore matching. The other method we proposed is to match the pores by ridges. In the first step, a fast ridge match method is proposed to find the corresponding ridges. Then the pores on corresponding ridges are matched using dynamic programming method. At last, a hierarchical fusion method is proposed for fingerprint matching, which integrates the match results of orient, minutiae, and pores.

Finally, some experiments are performed on the high resolution fingerprint database, which is downloaded from the Hong Kong Polytechnic University site. The results show that those two methods we proposed can achieve much higher recognition accuracy compare with the similar algorithm already exist.

Keywords: fingerprint identification, pore matching, division and conquer, fusion

目 录

摘 要.....	I
Abstract	II
第 1 章 绪 论	1
1.1 生物特征识别概述	1
1.2 高分辨率指纹的研究现状.....	2
1.2.1 高分辨率指纹的研究意义.....	2
1.2.2 现有高分辨率指纹识别的研究	4
1.2.3 高分辨率指纹的问题	5
1.3 本文主要工作及结构安排.....	5
第 2 章 基于细节节点的指纹匹配	7
2.1 引言	7
2.2 细节节点的提取	7
2.2.1 图像分割.....	8
2.2.2 归一化	9
2.2.3 方向场的计算.....	9
2.2.4 指纹的增强.....	10
2.2.5 二值化	12
2.2.6 细化	12
2.2.7 细节节点的提取.....	13
2.2.8 去伪细节节点.....	14
2.3 细节节点的匹配	15
2.3.1 细节节点的配准.....	15
2.3.2 匹配	15
2.4 本章小结.....	16
第 3 章 基于分块的汗孔匹配	17
3.1 引言	17
3.2 指纹特征的提取	17
3.3 指纹匹配.....	18
3.3.1 指纹细节节点和方向场的匹配.....	18
3.3.2 基于三角剖分分块的汗孔匹配.....	19
3.3.3 算法的时间复杂度	24

3.4 多特征的融合	24
3.5 本章小结	27
第 4 章 基于脊线的汗孔匹配	29
4.1 引言	29
4.2 前期处理	30
4.2.1 特征的提取	30
4.2.2 脊线的表示	31
4.2.3 脊线之间的关系	32
4.3 脊线的匹配	32
4.3.1 脊线相似度	33
4.3.2 脊线的重新配准估计	33
4.3.3 匹配	33
4.4 基于脊线的汗孔匹配	36
4.5 多特征的融合	38
4.6 本章小结	40
第 5 章 实验结果与分析	41
5.1 引言	41
5.2 实验数据库	41
5.3 性能测试	41
5.3.1 多特征融合测试	41
5.3.2 同类算法对比	43
5.3.3 效率对比	46
5.3.4 配准效果对比	47
5.4 本章小结	48
结 论	49
参考文献	50
哈尔滨工业大学学位论文原创性声明及使用授权说明	54
致 谢	55

第1章 绪 论

1.1 生物特征识别概述

生物特征识别是一种利用人的生物特征对身份进行认证识别的科学。与其他的方法不同，生物特征识别利用的是我们所拥有的个体生理特征，人的生物特征是最唯一的，而且就在分布在我们身体的各个部位，使用方便，避免了一个大活人需要一大堆文件来证明自己身份的尴尬。生物特征识别的目标就是将这些与生俱来的生理特征抽象量化，并进行计算比较，以达到最终识别分类的目的。生物匹配的流程可以大体分为：样本获取、提取特征、特征匹配三个步骤。在这一系列过程中，人类的唯一性生理特征会被提取出来，并转换为数字特征，然后与数据库中的所保存的样本数据进行对比，以实现人机系统交互，最终确定匹配或不匹配。

比尔盖茨曾说过：“生物特征识别技术利用个人的生理特征来识别个人身份，它将成为今后几年 IT 产业的一些重要革新。”实践表明，生物特征识别技术正在蓬勃发展，而且未来将会有更广阔的发展空间。

生物特征分为生理特征和行为特征两种，如图 1-1 所示。比如指纹，虹膜，掌纹，人脸，气味等都是生理特征；比如签名，语音，行走姿势等都为行为特征。在所有的生物特征中，指纹相对稳定，指纹作为人类生物特征识别的手段已经有很长的历史。1788 年梅耶第一次提出没有两个人的指纹会完全相同；1892 年英国人 Galton^[1]在对指纹进行了深入而完整的研究后，首次提出了指纹的唯一性；1899 年 Henry^[2]提出的指纹细节特征识别理论为现代指纹学奠定了最终的基础。由于其出色的唯一性和稳定性，从被发现时起，指纹识别技术就被广泛地应用于民用，刑侦等领域。在刑事领域，指纹更被称为“物证之首”。由于早期的技术限制，当时的指纹识别的速度慢，效率低，影响了指纹识别的普及。到了二十世纪六十年代，自动指纹识别系统（简称 AFIS）出现了，因为得到了计算机的帮助，指纹识别的速度和效率大大提高，指纹识别应用开始迅速发展。当然指纹识别也有缺点，指纹采集是侵犯性的，所以在应用时具有一定的局限性。从具体的使用情况来看，由于相对其他生物识别技术更加稳定，指纹识别使用比较方便，占据了更大的优势，更易为大众所接受。所以指纹识别技术占据了大部分的国内生物特征识别市场，被广泛应用到了许多方面。当然很多生物特征识别技术比如虹膜、人脸、视网膜、掌纹等识别技术也都得到了广泛的应用。但是由于这几种特征都有一定的

局限性并且不够成熟，所以相比指纹识别的发展还有一定的差距。

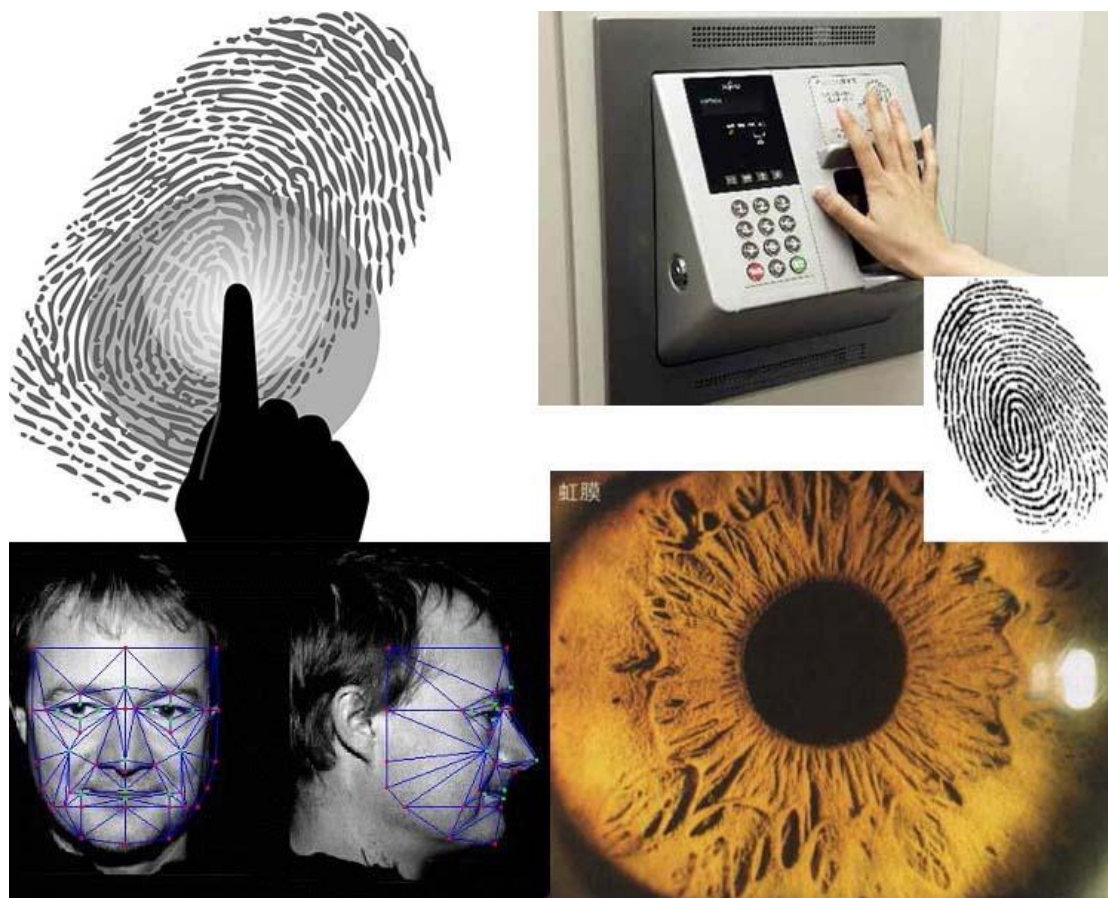


图 1-1 常用的生物特征识别技术

总之，生物特征识别现在已经开始被大众认同，并开始在社会的各个方面应用。而且生物特征识别更加安全，可靠。其中指纹识别技术是所有生物特征识别技术中发展的最为突出的。目前我国使用的第二代身份证中已经预留了指纹信息存储区，并在近期规定以后的身份证办理需要登记指纹信息。鉴于我国人口数量众多，可以说指纹识别的发展空间巨大。

1.2 高分辨率指纹的研究现状

1.2.1 高分辨率指纹的研究意义

自动指纹识别系统（automated fingerprint identification system，简称 AFIS）的应用前景变得越来越广泛。自动指纹识别系统被广泛应用于：刑侦，信息安全，医疗，公共安全，金融安全，社会福利等方面。特别是随着采集设备精度的提高，如图 1-2 所示，高分辨率指纹识别逐渐走进我们的视野。随着信息技术的发展，自

动化生物特征识别在人类生活中的应用越来越广泛，对识别的准确性的要求也越来越高，所以指纹识别的研究越来越深入。

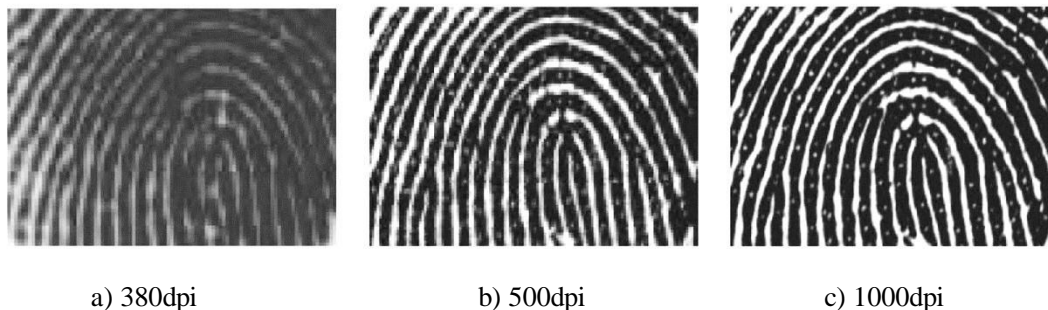


图1-2 不同分辨率下的指纹^[3]

高分辨率指纹是指分辨率大于等于 1000dpi (dots per inch) 的指纹图像。相比传统的指纹图像 (小于等于 500dpi 的指纹图像)，高分辨率指纹图像能够提取出更多的指纹特征信息，比如汗孔如图 1-3 所示，褶皱等第三层指纹特征信息，而传统的指纹识别^[4]只能识别第一层 (宏观指纹特征) 和第二层指纹特征 (Galton 特征和细节点)。统计显示，第一层指纹特征并不具有唯一性，但可以用来进行指纹分类；第二层指纹特征在指纹识别时具有一定的识别力，目前的大多数指纹识别都是基于第二层指纹特征。与第二层指纹特征相同，第三层指纹特征同样被刑侦领域的专家所认可，它同样具有唯一性和永久性，相比第二层指纹特征，信息量更多。因此在扩充了第三层指纹信息后，高分辨率指纹识别可以比传统的指纹识别提供更高的准确率和更好的鲁棒性，另外由于高分辨率指纹能够提取很多的信息，所以只需要一小片指纹残片就能实现识别，还能预防指纹的造假。因此高分辨率指纹识别吸引了大量的学者进行研究。

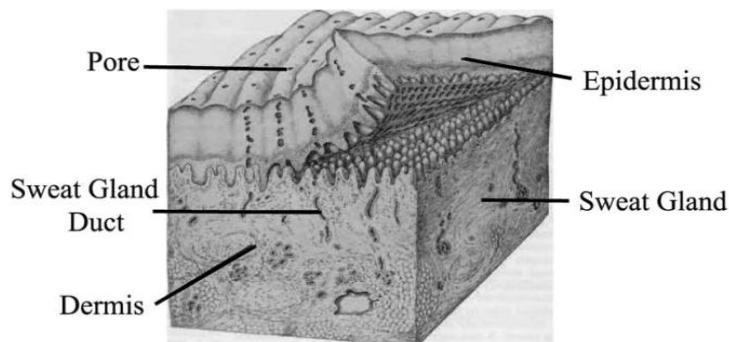


图 1-3 手指的皮层结构^[3]

高分辨率指纹识别的研究大体可以分两步，即指纹特征提取和指纹匹配。其中指纹匹配是指纹识别系统的核心内容之一，它是自动指纹识别的最后一步，也是非常关键的一步，对于整个高分辨率指纹识别系统的性能有很大的影响。指纹

匹配要解决的是对比从两幅指纹图像中提取出来的特征信息，进而判断这两幅指纹图像是否出自同一个指头。如何根据指纹的高精度信息进行匹配，是本文研究的重点。

与传统的指纹匹配相同，高分辨率指纹匹配面对的主要问题同样是旋转，位移，噪声和形变，误提取信息等问题。因为每次采集指纹时，人手不可能做到在同一位置，所以每次采集的指纹都会有一定的旋转、位移。而每次采集的角度和按压力度不同同样会造成指纹的形变，对于高分辨率指纹识别来说这种形变对于匹配结果的影响更大。另外高分辨率指纹提取出了大量的信息，这也对指纹的匹配造成了很大压力，大量的信息匹配需要大量的时间和空间。有效降低计算量是高分辨率指纹匹配需重点解决的问题之一。目前对于高分辨率指纹匹配的研究相对较少，大体来说，基于汗孔的匹配方法可以分为两类，第一类方法基于其他特征（比如细节点）配准的实现；第二类匹配方法是直接的汗孔匹配，这类方法实现基于局部的汗孔特征和整体变换实现。目前直接应用第三层指纹信息（比如汗孔）进行匹配的算法不是很多。相比之下，直接匹配算法比第一类算法更有效率，但是目前的直接汗孔匹配算法也有很多问题，每个汗孔只能与一个汗孔进行匹配，对于噪声十分敏感，需要高质量的指纹图像，且容易造成错误匹配等。

1.2.2 现有高分辨率指纹识别的研究

指纹匹配方法大多采用基于图形的匹配^[5-7]。图形匹配的实质是对纹线和特征点的匹配。它的原理是尝试所有的角度和位移把两个细节点集中相对应的点对应在一起。这类方法可以有效解决位移和旋转角度带来的匹配问题，而且在一定程度上能够解决噪声和图像模糊造成的误差。除了细节点本身的类型、数量和位置信息以外，指纹奇异点的类型和位置、细节点连线穿过的纹线数等信息在这类方法中都可以得到应用。这类指纹匹配方法被广泛应用于目前绝大多数自动指纹识别系统中。

在大多数情况下，点匹配方法都是很有用的，但是在实际应用中，其仍需改进，因为它存在如下两个方面的问题：一是需要对每个点进行比较匹配，故速度比较慢；二是对指纹图像的质量要求较高。碰到低质量的指纹图像时，可能会提取到很多错误的指纹特征信息，匹配效果会受到较大影响。

也有人提出了基于人工神经网络的匹配^[8-10]。人工神经网络有抗模糊能力强，容错性等优点，但这种方法首先需要进行样本训练，然后根据训练结果进行匹配。由于训练样本需要大量的时间，所以这种方法实时性差，不适合在线的指纹识别系统。

关于汗孔的匹配算法，目前都使用点模式的匹配。根据匹配前配准方法的不同，大体分为两类：第一类方法在匹配前，先要用细节点等信息进行配准；第二类匹配方法则是直接进行汗孔的匹配。关于汗孔的配准，目前已经提出了很多种方法，Kryszczuk^[11]等人提出了基于图像相关性的方法进行配准，然后再用几何距离的原则进行汗孔的匹配。这种方法有两个缺点：一是要获取最好的配准效果，需要大量的时间和空间，因为为了配准需要遍历所有可能的旋转角度和平移的距离来寻找测试图像和模板图像的重叠区域，这使得运算量极大；二是最后的识别准确率严重依赖配准的准确率，如果碰到噪声或非线性形变则对匹配的结果影响很大，即使是微弱的噪声也会对配准结果产生极大的影响。Anil. K. Jain^[3,12]等人提出了另一种方法，基于细节点的配准算法，这种方法首先应用细节点进行配准，然后应用 ICP 算法对配准细节点对的长方形邻域的汗孔进行匹配。这种方法更加有效，但是这种方法也有缺陷，它需要大量的细节点来进行配准，而且只考虑到了细节点邻域的一小块领域的汗孔。第二类高分辨率指纹匹配直接应用汗孔进行匹配而不需要配准。Zhao^[13-15]等人提出了一种分层汗孔匹配方法。这种方法首先根据汗孔附近区域相关性找到粗略的对应匹配，然后再根据这一结果用一种转换模型进行再匹配。这种算法的优点是对于不稳定的汗孔提取有很高的鲁棒性。但是它还是有一些局限性：首先用汗孔点附近的区域的相关性进行粗略匹配准确率不高，并不能保证汗孔能够匹配到正确的汗孔，比如平行脊线上的汗孔，严重的形变或噪声都可能与正在匹配汗孔有很高的相关性，所以在一对一匹配时选择相似性最高的区域不一定是正确的；另外，在第一步大体匹配的阶段并不能保证整体的可靠性，每一对匹配汗孔的可靠性都不一样。这些区域可靠性的不同也间接表现了图像形变的情况，而该算法没有利用这些信息。

1.2.3 高分辨率指纹的问题

总的来说，现在的高分辨率指纹匹配算法的准确率相对不高，计算复杂度很高，对抗噪声能力差，高分辨率匹配的方式相对比较单一。结合这些问题，本文提出了两种算法：基于分块的汗孔匹配算法，基于脊线的汗孔匹配算法，这两种算法有效降低了汗孔匹配的计算复杂度，对于非线性形变处理较好，在结合细节点，方向场等信息融合后，准确率较同类算法亦有所提高。

1.3 本文主要工作及结构安排

本文对高分辨率指纹的匹配做了深入的研究，提出了两种基于分治思想的高分辨率指纹匹配算法，主要工作如下：

(1) 采用分治的思想, 提出了基于细节点对指纹汗孔分块的匹配算法, 算法首先根据细节点的匹配点对指纹三角剖分, 然后找到每个块中的汗孔分别计算匹配得分, 最终融合细节点, 汗孔和方向场三种特征得到最终匹配结果, 得到了较好的效果。

(2) 提出了基于脊线匹配的汗孔匹配算法, 该算法同样基于分而治之的思想实现。算法首先根据脊线进行匹配, 然后根据找到的匹配脊线, 计算匹配脊线上的汗孔匹配得分, 然后融合细节点和方向场的匹配结果, 得到了较好的匹配效果。

本文的结构安排如下:

第一章绪论, 简单介绍了生物特征识别的现状, 高分辨率指纹识别的研究目的与意义。

第二章基于细节点的指纹匹配, 介绍了基于细节点的高分辨率指纹匹配算法。

第三章基于分块的汗孔匹配, 提出了一种基于细节点三角剖分, 对汗孔匹配, 然后融合细节点, 方向场的高分辨率指纹匹配算法。

第四章基于脊线的汗孔匹配, 提出了一种基于脊线匹配汗孔再融合细节点和方向场的高分辨率匹配算法, 介绍了算法的原理。

第五章实验结果和分析, 对本文提出的两种算法进行了实验测试和分析, 并对未来的发展做出了展望。

第2章 基于细节点的指纹匹配

2.1 引言

因为本文中提出的两个算法都用到了基于细节点的指纹匹配，并且在最终的实验中也要与基于细节点的指纹匹配进行对比，所以我们首先介绍一下基于细节点的指纹匹配的实现过程。

本章中实现的指纹识别算法是基于冯建江的方法^[16]实现。在指纹增强阶段，采用 Gabor 滤波的方式，根据计算得到的方向场和频率，对指纹进行了增强；在匹配阶段，采取基于细节点距离的原则，进行指纹的匹配。本章数据库采用香港理工大学高分辨指纹特征库，并用本章的算法进行了测试，得到了较好的效果。

2.2 细节点的提取

指纹的细节点是指纹的第二层特征信息，是目前指纹识别主要应用的指纹特征，主要包括终结点，分叉点这两种类型。细节点的提取如图 2-1 所示，主要分为几个步骤：指纹图像的分割，指纹的归一化，计算图像的方向场和频率，指纹的增强，指纹的二值化，指纹的细化，细节点的提取，去除伪细节点。

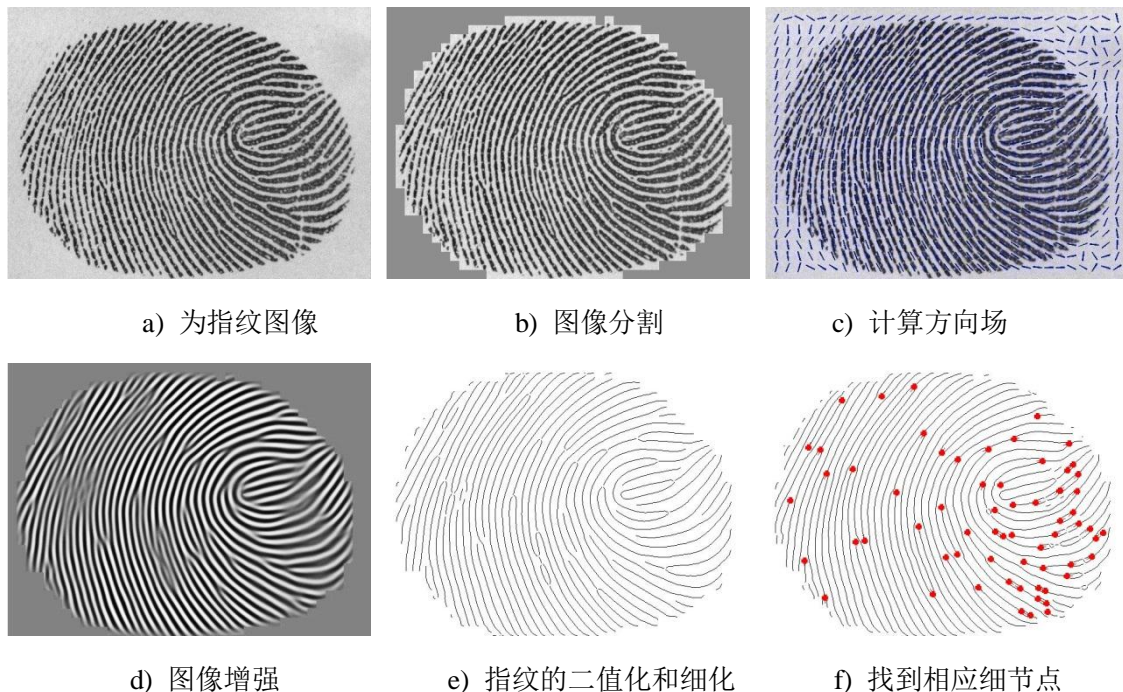


图 2-1 提取细节点的几个步骤

2.2.1 图像分割

由于指纹采集设备的原因，采集到的指纹图像，包含了一些不包含指纹的区域和噪声严重的区域，我们称之为背景，其中我们把有脊线的清晰区域称为感兴趣区域（ROI）。如图 2-2 所示，在指纹提取的第一步就是对指纹图像进行分割，找到指纹的感兴趣区域。找到感兴趣区域的同时，去除背景中的错误信息，这样做能够有效减少特征提取所需的处理时间及误提取率。本章采用计算灰度平均值和方差的方式寻找指纹图像中的感兴趣区域，原理在于 ROI 区域包含大量的脊线，脊峰和脊谷的明暗分别大，致使其区域的灰度方差也较大，分析背景区域，由于没有脊线或者图像模糊，所以灰度方差相对较小。那么我们就可以利用这个特点，计算每个指纹区域的方差，进而对指纹图像进行分割。指纹图像分割的大体流程如下：

- (1) 首先将指纹分成 ω 宽 ω 长的若干个小块。
- (2) 分别计算每个小块内的像素平均值。由公式 (2-1) 求得：

$$Mean(m,n) = \frac{1}{\omega \times \omega} \sum_{i=1}^{\omega} \sum_{j=1}^{\omega} image(i,j) \quad (2-1)$$

式中 $image(i,j)$ ——指纹图像每个小块中第 i 行第 j 列像素点的像素值；

$Mean(m,n)$ ——指纹图像第 m 行第 n 列的子块的像素平均值；

m ——图像的高度 H 除以块的高度 ω ；

n ——图像的宽度 W 除以块的宽度 ω 。

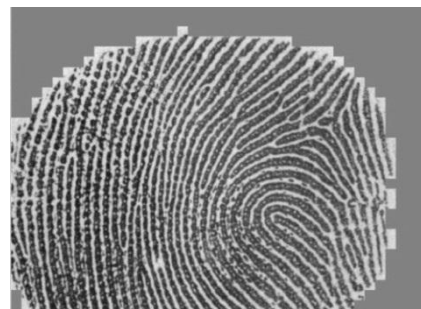
- (3) 然后我们根据块的像素平均值计算相应的灰度值均方差。见公式 (2-2)：

$$Var(m,n) = \frac{1}{\omega \times \omega} \sum_{i=1}^{\omega} \sum_{j=1}^{\omega} (image(i,j) - Mean(m,n))^2 \quad (2-2)$$

- (4) 根据得到的块灰度值均方差，当 $Var(m,n)$ 小于阈值 T 时（本章选取阈值 T 为 0.3），则认为这块图像为背景；否则，我们认为这块图像为 ROI 区域。



a) 指纹分割前



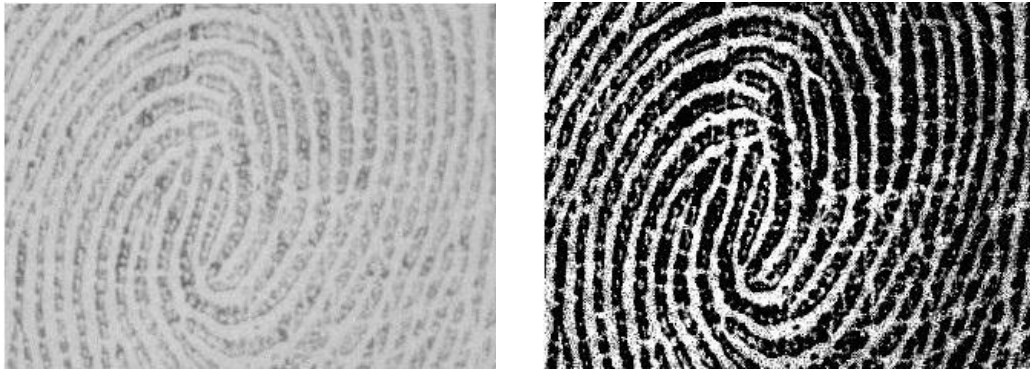
b) 指纹分割后

图 2-2 指纹图像的分割

2.2.2 归一化

受到采集环境的影响，指纹采集到的图像会出现整体偏黑或者偏白的情况，使指纹的脊线部分和脊谷部分区分度不大。这对于后期指纹的增强和后期提取不利，所以我们采用归一化的方法对指纹进行处理以得到区分度较好的脊峰和脊谷图像。如图 2-3 所示，通过归一化后指纹的脊峰和脊谷之间明暗变化明显变得平缓，使图像均值和方差更接近于理想状态，而且使用归一化后图像的清晰度不会降低。本章使用的归一化方法如公式（2-3）所示：

$$Normal(i, j) = \begin{cases} M_0 + \sqrt{Var_0 (image(i, j) - Mean)^2 / Var}, & image(i, j) > Mean \\ M_0 - \sqrt{Var_0 (image(i, j) - Mean)^2 / Var}, & other \end{cases} \quad (2-3)$$



a) 归一化前

b) 归一化后

图 2-3 指纹图像的归一化

2.2.3 方向场的计算

指纹方向场是指图像上一点对应局部脊线的方向值，真实表现了脊线走势和形状，是对指纹的一种抽象表示。方向场被用来进行指纹图像的增强。最早图像分割也用到了方向图；在指纹图像增强阶段，为了使脊线更加突出，我们需要根据脊线的方向来进行指纹的图像处理；还可以根据方向图对脊线平滑处理，使指纹中采集时断裂的脊线连接起来；在指纹提取阶段也要用到方向场。指纹脊线方向场的提取是指纹提取阶段的重要一环。

本章中方向场的提取采用梯度法求得。梯度法是目前指纹方向场提取应用的最多的方法，其原理是首先计算图像点的梯度，因为方向场的方向与该点梯度的方向垂直，所以只要计算该点的梯度就能求出方向场的值。方向场的计算结果如图 2-4 所示。方向场的主要计算流程如下：

- (1) 首先将指纹分成 ω 宽 ω 长的若干个区域；
- (2) 接下来取每个区域的每个像素点，分别从水平和竖直两个方向计算相应的梯度值，对于每个图像点，求得其梯度向量 $[G_x(x, y), G_y(x, y)]^T$ ；
- (3) 然后计算每一个区域的平方梯度向量的平均值，梯度 ϕ 的计算方法如公式 (2-4) 所示，脊线方向场 θ 计算如公式 (2-5) 所示：

$$\phi = \begin{cases} \frac{1}{2} \tan^{-1} \left((G_{xx} - G_{yy}) / (2 \times G_{xy}) \right), G_{xy} \geq 0 \\ \frac{1}{2} \tan^{-1} \left((G_{xx} - G_{yy}) / (2 \times G_{xy}) \right) + \pi, G_{xy} < 0, G_{xx} - G_{yy} \geq 0 \\ \frac{1}{2} \tan^{-1} \left((G_{xx} - G_{yy}) / (2 \times G_{xy}) \right) - \pi, G_{xy} < 0, G_{xx} - G_{yy} < 0 \end{cases} \quad (2-4)$$

$$\theta = \begin{cases} \phi + \frac{\pi}{2}, \phi \leq 0 \\ \phi - \frac{\pi}{2}, \phi > 0 \end{cases} \quad (2-5)$$

式中 G_{xx} ——每个块内所有点 $G_x \times G_x$ 的和；

G_{yy} ——每个块内所有点 $G_y \times G_y$ 的和；

G_{xy} ——每个块内所有点 $G_x \times G_y$ 的和。

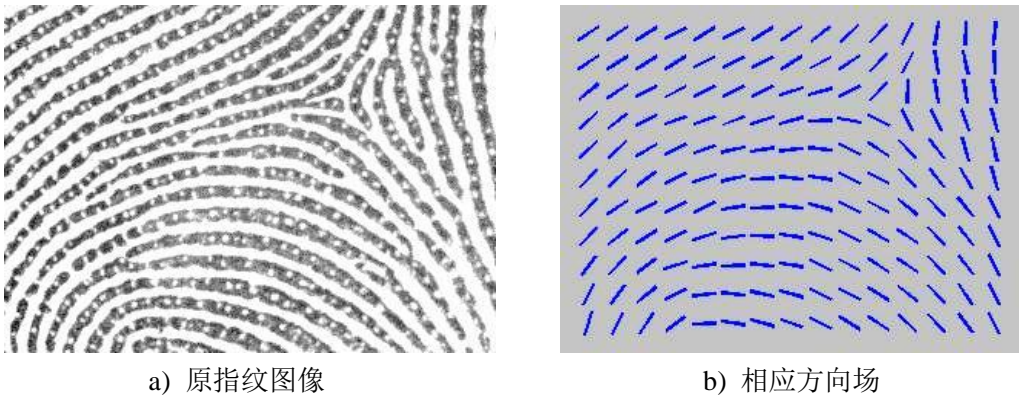


图 2-4 指纹的方向场计算

2.2.4 指纹的增强

由于受到采集仪器所处环境和手指当时状态的影响，我们采集到的指纹通常都会伴随着各种噪声。噪声分两类，第一类噪声具有一定的规律性，属于系统误差比较容易恢复；第二类噪声相对复杂，由于每个人得手指情况各不相同，造成噪声的原因也相对复杂，这类噪声较难恢复。指纹增强对于指纹识别系统十分重要，这一步骤效果的好坏直接影响整个指纹识别系统的性能。指纹图像是包括脊

峰和脊谷两种纹线，这些纹线中包含了大量的特征信息。在指纹的采集过程中通常伴随着各种各样的噪声，所以对于采集到的指纹需要进行增强。在一个小区域内，脊峰和脊谷可以被看做是一段正弦波，像正常的波形一样具有频率和方向。指纹增强是对指纹图像中模糊的区域进行加强，加强脊峰和脊谷的区别，以提高后续指纹提取的准确度和可靠度。目前指纹增强的方法很多，Maio 等人提出的直接跟踪脊线方法^[18-20]，Almansa 提出的自适应指纹脊线增强算法^[21]，Gorman 的多分辨技术增强算法等^[22]，也有提出在频域范围内做增强的^[23,24]，本文选择采用 Gabor 的方法进行图像的增强。Gabor 滤波是图像增强算法中最常见的一种，是一种基于方向场滤波的增强算法^[25,26]。首先我们假设一个小的区域内的指纹是具有一定频率的平行线，然后根据这一区域的平行线我们建立数学模型，接下来我们使用 Gabor 窗沿着脊线进行过滤，使用这种方法我们就实现了指纹的增强。这种方法在顺着脊线方向上具有平滑的作用，能够修复一些断裂的脊线，同时具有良好的频率可选性，能够在有效去除噪声的同时，保持脊线的完整性。

本章采用的 Gabor 滤波器定义如公式（2-6）所示：

$$\begin{cases} h(x, y, \theta, f) = \exp\left(-1/2 \times \left(x_\theta^2 / \sigma_x^2 + y_\theta^2 / \sigma_y^2\right)\right) \times \exp(j \times 2\pi \times f \times x_\theta) \\ x_\theta = x \times \cos \theta + y \times \sin \theta \\ y_\theta = -x \times \sin \theta + y \times \cos \theta \end{cases} \quad (2-6)$$

式中 x_θ ——沿着脊线方向上像素点对应的坐标；

y_θ ——垂直脊线方向上像素点对应的坐标；

θ ——滤波器的中心方向；

f ——滤波器中心频率；

σ_x^2 ——频率通带大小；

σ_y^2 ——方向通带大小。

图像增强的效果如图 2-5 所示，基于 Gabor 滤波器的指纹图像增强方法，分为三个步骤：

- (1) 首先根据上一节的方法计算指纹图像所有点的方向场 $\theta(i, j)$ ；
- (2) 接下来每个点邻域内的脊线频率 $f(i, j)$ ；
- (3) 利用公式（2-7）进行滤波，

$$image'(i, j) = \sum_{u=-\omega_g/2}^{\omega_g/2} \sum_{v=-\omega_g/2}^{\omega_g/2} h(u, v, \theta(i, j), f(i, j)) \times image(i, j) \quad (2-7)$$

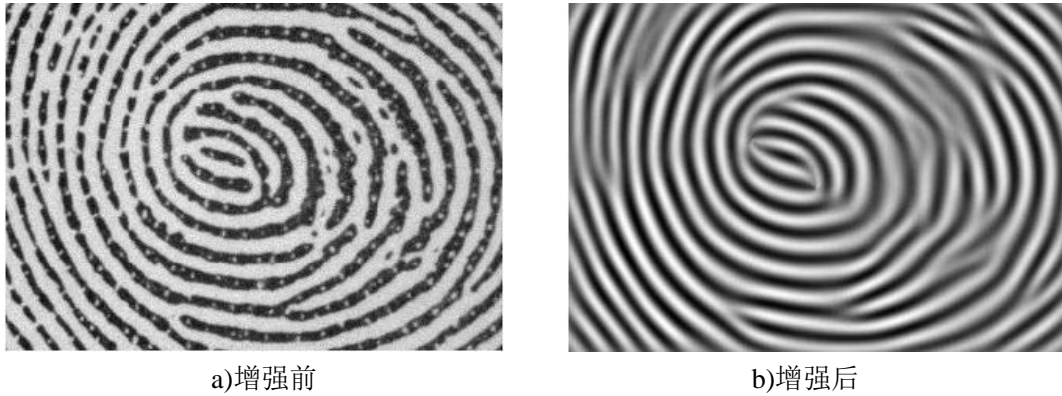


图 2-5 指纹的增强

2.2.5 二值化

在得到增强后的图像后，我们接下来进行指纹的二值化处理，把图像转换成二值图像。这样就能够使指纹图像的脊线和谷线的对比度更加明显，便于后续处理。本章二值化采用的是像素值对比的方法，首先设定一个阈值，然后扫描增强后的指纹图像上的每一个像素点，如果大于阈值则设为 1，如果小于则设为 0，这种方法简单有效，可以快速实现图像的二值化，采用的计算方法如公式（2-8）所示。指纹二值化后的效果如图 2-6 所示。

$$G(i, j) = \begin{cases} 1, & \text{image}(i, j) > T \\ 0, & \text{image}(i, j) < T \end{cases} \quad (2-8)$$



图 2-6 指纹的二值化

2.2.6 细化

在得到二值化的图像后，我们就可以对图像进行细化。细化的目的是在保持原来形状的条件下，将脊线处理成宽度为一个像素的线条。目前的细化算法很多，大体可分为：串行、并行和混合三类。串行细化算法采取逐个像素点细化的方式，

这种方法细化效果好，速度慢；并行细化算法采用的是同时对一批像素点进行细化的方式，这种方法速度快，但效果较差；混合细化则是结合了串行和并行两种方式的细化，这种方法结合了两种细化方法的优点，速度快，细化效果好。

本章的细化方式是采用形态学方法，根据每个点周围邻域的不同情况进行细化，效果如图 2-7 所示。

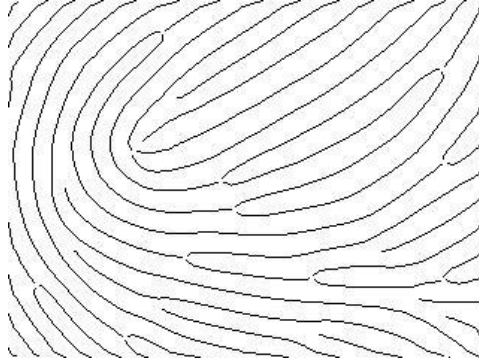


图 2-7 指纹的细化

2.2.7 细节节点的提取

细节节点的提取方法有很多^[27-29]，本章采用 8 邻域像素变化计数法进行指纹细节节点的提取。在得到细化后的图像后，指纹图像变成由 0,1 组成的黑白图像。其中 0 为白色的背景点，1 为黑色的脊线点。接下来我们使用一个三乘三的窗口扫描所有细化后的脊线点，来判断该点是否是细节节点，并判断细节节点的类型。检测的方法是扫描每个像素点的 8 邻域，如图 2-8 所示，计算 8 邻域的像素变化次数。扫描每个点的窗口定义如公式 (2-9) 所示：

$$cn(p) = \frac{1}{2} \sum_{i=1..8} |val(P_{i\%8}) - val(P_{i-1})| \quad (2-9)$$

式中 $cn(p)$ ——8 邻域的像素点变化数的一半；

$val(P_i)$ ——像素 8 邻域第 i 个点的像素值。

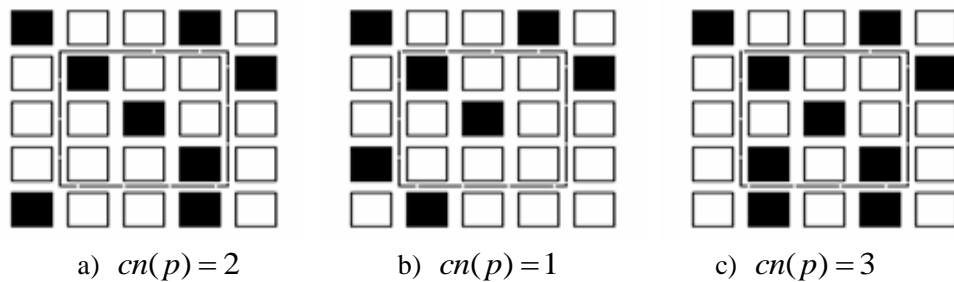


图 2-8 脊线上的点的 8 邻域的三种情况^[4]

扫描时按照顺时针的方向计算两个点的像素值，如果不同则计为变化一次。分析细化后指纹图像，脊线上的点的 8 邻域的变化次数只有 3 种情况，

(1) $cn(p)=2$ ，这时说明像素点 8 邻域中像素变换了 4 次，则这个点成为脊线上的点；

(2) $cn(p)=1$ ，这说明像素点的 8 邻域中像素变换了 2 次，这时我们称这个点为脊线终结点；

(3) $cn(p)=3$ ，这说明像素中心点的 8 邻域像素变换了 6 次，这时我们称这个点为分叉点。

2.2.8 去伪细节点

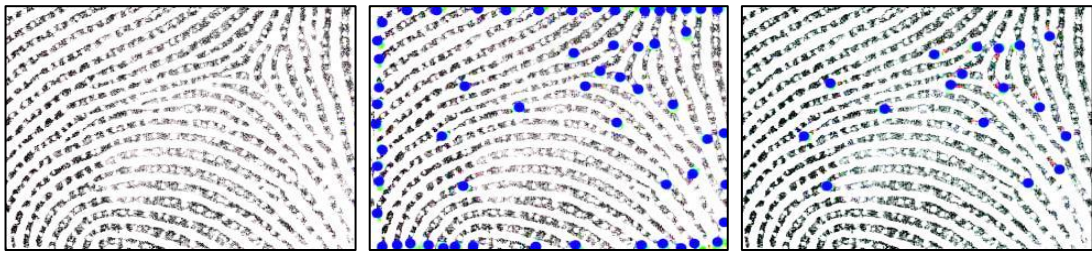
伪细节点的存在会造成指纹识别系统整体性能的下降，因此我们要把这些点去除掉。由于现实情况的复杂性，比如图像的质量问题，噪声干扰等都会造成伪细节点的产生。另外由于算法的局限性，在提取特征时，二值化和细化等步骤都会产生伪细节点。在去除伪细节点我们也要注意一些问题：首先在去除伪细节点后，要保留下正确的细节点；计算复杂度要低；鲁棒性要强，能够处理不同类型的伪细节点；在处理过程中不能产生新的伪细节点。本章根据伪细节点类型的不同，我们分别采用如下三种方式进行去除，去除后的效果如图 2-9 所示：

(1) 首先去除图像边缘的伪细节点：这类伪细节点大多位于图像的边缘，产生的原因是采集环境影响和特征提取误差造成的。这种伪细节点数量巨大，对最终的指纹匹配结果影响很大。在指纹分割阶段，我们已经将指纹图像分割成为若干个小的区域，为了去除边缘伪细节点，我们扫描所有指纹区域，对于处于背景区域邻域范围内提取出来的细节点，我们都视为伪细节点去除，使用这种方法，我们就能有效去除边缘伪细节点。

(2) 接下来去除噪声和模糊图像造成的伪细节点。这类伪细节点种类繁多，令人头痛，其中噪声造成的伪细节点尤为难处理，毛刺就是其中之一。毛刺产生后，形状与短脊线类似，这会造成两种伪细节点：终结点和分叉点。这种伪特征有一个公共的特点，那就是它们之间的距离都较短。为了去除这类伪细节点，我们取每个细节点与所有细节点进行比较，找到与本细节点距离小于 R 的所有细节点，如果找到了，说明这两个细节点都是由于毛刺而被误提取出来的，这两个细节点都是伪细节点，我们把两个点都删掉。为了去除模糊图像造成的伪细节点，我们同样扫描每一对细节点，当它们的距离小于阈值 T 时，就认为这两个细节点是由于图像模糊造成的，两个点都删掉。

(3) 最后我们去除由于错误连接造成的伪细节点。受到环境和指纹状态的影

响，指纹图像在采集时无法得到最真实的指纹图像，指纹上的脊线会被汗水，污迹等连在一起，形成很多本来没有的脊线，或者把本不相交的脊线错误的连在一起，最终形成了分叉或者环形，这时就会提取出一些本不存在的分叉点等伪细节点。基于这种情况提取出的细节点，大多是两个距离很近的分叉点，而且两个点的连线与其所在脊线的方向一致。从这一角度出发，为了去除这类伪细节点。我们逐个扫描每一对细节点对，如果这两个点距离小于一个阈值 T ，并且两个细节点的连线与两个点所在脊线的方向场处于同一水平线上，则认为这两个点都是由错误连接而被误提取出来的，两个点都删掉。



a) 原指纹图像

b) 初步提取的细节点

c) 去伪细节点后

图 2-9 去除伪细节点

2.3 细节点的匹配

2.3.1 细节点的配准

在高精度指纹识别系统各模块间的关系中，指纹图像的配准位于寻找匹配特征点对之前，是匹配中关键的一步。目的是找出对准两个指纹图像和特征所需的平移和旋转参数。本章的配准策略采用边配准边匹配的方法，首先找到主匹配点对，然后根据相应坐标和方向场得到平移和旋转参数，然后根据这些参数对其他点进行配准，计算匹配得分，匹配算法最终要扫描所有可能的主匹配点对，找到匹配得分最高的作为最终的结果。

2.3.2 匹配

本章中细节点的匹配采用点模式匹配。输入指纹细节点特征保存如公式(2-10)：

$$A = \left((x_1, y_1, \theta_1)^T, \dots, (x_m, y_m, \theta_m)^T \right) \quad (2-10)$$

模板指纹细节点特征保存如公式 (2-11)：

$$B = \left((x_1, y_1, \theta_1)^T, \dots, (x_n, y_n, \theta_n)^T \right) \quad (2-11)$$

分别包含了细节点的坐标和细节点的方向。匹配第一步首先计算所有输入指纹细

节点和模板指纹细节点的相似度，相似度的计算定义为每对细节点周围半径 R 到半径 $R+l$ 的范围内的细节点的匹配度，在得到所有细节点之间的相似对后，根据相似度得分进行排序；第二步，选择相似度较高的细节点对，作为主匹配点对，根据其坐标值和方向得出转换角度和位移。再根据转换角度和位移对细节点进行校准。第三步，对校准的细节点对使用基于距离原则的方法进行匹配，对于两个距离小于阈值 d 的细节点，我们就认为其匹配。由于存在非线性形变等因素的影响，对于距离阈值的选择，我们使用有弹性的阈值选择。对于距离主匹配对较远的点，我们适当放松距离阈值 d ，使其能够允许非线性形变造成的偏移。根据距离的远近，我们选择三种阈值，根据与主匹配对的距离按比例进行缩放。其中为了得到准确的结果，分别选择相似度较高的 10 对细节点作为主匹配对，然后计算相应的匹配结果，最终取最好的结果作为匹配得分。匹配得分的计算如公式 (2-12)：

$$M = \frac{2 \times N_m}{N_t + N_q} \quad (2-12)$$

式中 N_m ——匹配的细节点个数；

N_t ——模板指纹的细节点个数；

N_q ——输入指纹的细节点个数。

2.4 本章小结

本章实现了在高分辨率下指纹的细节点提取与匹配算法，细节点的提取和匹配是本文中高分辨率指纹匹配的一部分，在基于分块的汗孔匹配和基于脊线的汗孔匹配都应用到了细节点的匹配结果。本章细节点的提取与匹配采用比较成熟的方法实现，指纹的增强我们采用的是 Gabor 滤波的增强方法；在匹配阶段我们根据基于距离的原则进行细节点的匹配。在采用香港理工大学的高分辨率指纹图像库测试后，本章采用的细节点提取和匹配算法效果良好，达到了一定的准确率。

第3章 基于分块的汗孔匹配

3.1 引言

根据目前已经提出的汗孔匹配算法，我们提出了一种基于分治思想的汗孔匹配算法。分而治之是一种我们很熟悉的策略，这种思想在很久以前就应用于我们的生活中，古老而实用。分治的思想是将一个较复杂的问题分解成若干个小问题，这样每个小问题相对来说都容易解决，以达到解决整个问题的目的。在高分辨指纹识别问题中，由于指纹图像的尺寸很大，每个区域的形变都各不相同，这对于指纹匹配带来了很大麻烦。采用分而治之的方法，把一个很大的指纹图像分割成若干个小的区域，分别根据区域的特征区别处理，能够很好地解决局部形变对高分辨率指纹识别的影响。本章主要介绍采用基于分块的汗孔匹配算法的具体实现过程，分别说明了采用细节点分块的方式，以及与其他指纹特征融合的策略。并在一个小规模的数据库上做了初步的实验，得到了较好的效果。

3.2 指纹特征的提取

指纹特征的提取，对于指纹识别的整体性能起着关键的作用。特征提取效果的好坏，是否能够去掉伪特征，提取到最准确的特征，对最终的指纹匹配影响巨大。本章中的指纹匹配算法用到了方向场，细节点，汗孔等信息，这些特征均采用本文第二章中介绍的方法提取。

汗孔随机分布于脊线上，并且位置不会随时间变化而移动^[30]。目前汗孔的提取方法有很多，Ray 等人^[31]提出的直接在灰度图像上提取汗孔的方法，Jain^[3,12]等人提出了一种提取汗孔的方法，这种方法利用小波变换来寻找汗孔所在位置。这两种方法有一定的优点，但也有不足，我们采用分块自适应的汗孔提取方法提取汗孔^[32]。提取步骤和提取细节点的方法类似：

- (1) 进行图像分割，选取指纹所在区域；
- (2) 计算指纹图像的方向场；
- (3) 构造两个高斯滤波器；
- (4) 分别对指纹图像进行两次高斯滤波，得到增强的汗孔和脊线；
- (5) 然后根据增强结果去掉伪汗孔得到提取的汗孔信息。

最终我们提取的汗孔特征为每个汗孔的坐标。

3.3 指纹匹配

指纹的特征匹配解决的是对两幅指纹进行对比，判断两个指纹是不是出自同一个手指。指纹的匹配是指纹识别的关键一步。对于指纹识别的性能起着关键的作用。

指纹匹配要解决的问题，主要是由于手指每次采集时与采集器接触状态不同等造成的。另外由于指纹提取算法等的因素，也造成了特征点提取不准确等影响。指纹匹配的算法就是要尽量减小平移，旋转，形变带来的影响，使指纹匹配的结果尽量准确。

首先要进行的就是指纹的配准，分析高分辨率指纹的特点，由于分辨率的提高，使指纹的细节被放大，但与此同时，局部的非线性形变也被放大。这些局部的形变对于汗孔这种第三层特征来说，影响很大。所以采用类似第二层指纹特征（细节点）的全局的匹配方法，已经不能适用于汗孔的匹配。本章提出一种采用分块配准，分块匹配的方法，用来解决这个问题。算法大体流程为，首先计算细节点和方向场的匹配；然后根据细节点匹配结果进行三角剖分，对汗孔进行分块匹配；最后对细节点，汗孔和方向场得分进行匹配级融合。

在多特征的融合后，算法的等误率有明显下降，相比每种特征单独匹配结果，都有明显的提高，相比同类算法，也有一定的优势。算法的第一步是细节点和方向场的匹配。以下首先介绍本章的细节点和方向场的匹配算法。

3.3.1 指纹细节点和方向场的匹配

本章中细节点的匹配采用的是上一章介绍的方法，既对比每对细节点的坐标和方向，其中细节点已经采取上一章的提取算法提取出来，其中输入指纹与模板指纹细节点特征的保存形式如公式（3-1）、（3-2）所示：

$$I = \left((x_1, y_1, \theta_1)^T, \dots, (x_p, y_p, \theta_p)^T \right) \quad (3-1)$$

$$M = \left((x_1, y_1, \theta_1)^T, \dots, (x_q, y_q, \theta_q)^T \right) \quad (3-2)$$

分别包含了细节点的坐标和方向。算法首先计算每一对细节点的相似度，相似度的计算定义为距离相应细节点距离 S 到 $S+L$ 的同心圆内包含的细节点的匹配度，接下来根据计算出来的匹配度进行细节点对的排序；然后我们选择相似度较高的细节点对作为主匹配点对，计算这对细节点的位移与方向场的差值，作为我们的配准参数。然后根据这一配准参数，对其他的细节点进行坐标校准。最后我们使用基于距离原则的方法搜索每对细节点，如果距离小于阈值 d ，认为其匹配。在搜

索每对细节节点是否匹配时，我们要根据不同的情况对阈值进行缩放，这样做的目的是为了解决非线性形变对匹配带来的影响。具体的做法是，我们根据与主匹配点对的距离对阈值进行选择。我们首先选择三个门限值及三个可选阈值，然后将与主匹配点对的距离与三个门限值比较，根据与三个门限值得关系和相应阈值，计算出最终判断两个点是否匹配所需的阈值。在匹配过程中，我们分别根据相似度较高的十对细节节点对作为主匹配对，并计算配准参数和坐标校准，然后基于距离的原则寻找匹配的细节点对，最后选择最高的匹配得分作为最终的得分。这样做的好处是不同尝试所有可能的主匹配对，可以在一定程度上减小配准所需复杂度。匹配得分的计算方法如公式（3-3）所示：

$$score = N_p / N_m \quad (3-3)$$

式中 N_p ——匹配的细节点个数；

N_m ——模板指纹和输入指纹的细节点个数的平均值。

接下来方向场的匹配方法采取计算每对相应像素点的方向场的差值的方法。首先尝试每种旋转角度和位移，并找到重叠区域，再计算这些区域每一对图像点的方向场的差值。如果方向场的大小相差小于阈值 ε ，则认为这两个点的方向场匹配，然后计算整个图像的匹配得分，匹配得分计算如公式（3-4）：

$$M_\theta = N_p / N_q \quad (3-4)$$

式中 N_p ——方向场匹配的像素点的个数；

N_q ——重叠区域的像素点的个数。

3.3.2 基于三角剖分分块的汗孔匹配

分析目前的汗孔匹配算法，大都采取整体匹配的方式，使用全部的汗孔进行类似细节节点的整体匹配，或者根据每个汗孔一对一的直接匹配。整体匹配在配准阶段会产生极高的时间复杂度。而直接匹配虽然没有使用所有汗孔进行配准，但是对每对汗孔都进行一次匹配计算，同样会产生较高的时间复杂度。分析高分辨率指纹图像，同一个指纹，在 1200dpi 的指纹采集器获得图像比在 500dpi 下获得的图像大出一倍多，那么在每个局部，指纹的局部形变也被扩大了，如果还采用原来的基于低分辨率的指纹匹配算法进行匹配就会遇到时间复杂度高，准确率低的问题。在本章根据对现有汗孔匹配算法的分析以及高分辨率指纹的特点提出分块匹配的方法。在分块后分别进行块内汗孔的坐标配准，这样可以大大降低配准的复杂度，并降低局部形变带来的误差。

因为目前细节节点的匹配算法已经相对比较成熟，而且相对准确，所以本章选择细节节点进行三角剖分。分块匹配算法的第一步首先根据细节节点进行匹配，然后根据匹配的细节点对进行三角剖分；第二步根据三角剖分后得到的三角形三个顶点（细节点）坐标进行配准参数的估计；第三步根据配准参数对三角形内的汗孔坐标进行校准；第四步比较校准后的汗孔坐标，基于距离原则找到匹配的汗孔；第五步统计所有三角形内的汗孔总数和匹配的汗孔总数，计算最后的汗孔匹配得分。本章算法流程如下：

- (1) 根据细节节点进行匹配，找到匹配细节点对；
- (2) 根据细节点对进行三角剖分；
- (3) 根据三角剖分结果找到重合的三角形；
- (4) 根据三角形顶点坐标，运用最小二乘法估计三角形配准参数；
- (5) 对三角形内的汗孔进行坐标校准；
- (6) 运用动态规划法对汗孔进行匹配；
- (7) 统计所有三角形匹配汗孔个数，和汗孔总数，计算最终匹配得分。

3.3.2.1 指纹的分块

对指纹的分块采用三角剖分的方式进行，首先根据细节节点进行匹配，再根据得到的匹配细节点对进行三角剖分，这样就实现了指纹的分块。因为目前细节节点的匹配算法比较成熟，所以选择根据细节节点的匹配结果进行三角剖分。三角剖分可以根据一个给定的不共线的点的集合把图像分割成若干个三角形，并且保证每个三角形都不重合，三角剖分由于其成熟性被广泛应用于数值分析以及图形学领域。本章选择 Delaunay 三角剖分，这种方法是使用最广泛的三角剖分方法之一^[33]。Delaunay 三角剖分具有：最接近、唯一性、最优性、最规则、区域性、具有凸多边形的外壳等优异特性。

根据细节节点进行三角剖分后，就将原来的指纹分成了若干个三角形，而且因为采用事先匹配好的细节点作为三角剖分的依据。所以在三角剖分后的三角形都是以细节点为顶点，而由于细节节点的提取是依赖与脊线的位置，汗孔又位于对应的脊线上，所以根据对应的细节点剖分成的三角形内的汗孔点，就是对应区域的汗孔。然后根据已知的三角形顶点坐标，估计配准参数就能很好的解决局部形变对指纹匹配的影响。

图 3-1 为根据细节节点三角剖分后的效果图。可以看到，其中很小一部分的三角剖分结果不同，原因是因为这些区域的细节节点的相对偏移较大，可以说这部分的形变较大，那么，我们选择三角剖分结果一样的三角形，作为汗孔匹配区域。这

样做的原因是指纹中汗孔的数量众多。汗孔所携带的信息量远远超过正确匹配所要的量，根据克日什科夫的结论^[11]，对于适当选择部分区域汗孔而不是全部汗孔进行匹配，并不会降低最终的匹配正确率。本章选择三角剖分结果一样的区域，可以减少形变较大区域对匹配结果的影响，同时也不会影响最终的匹配结果。

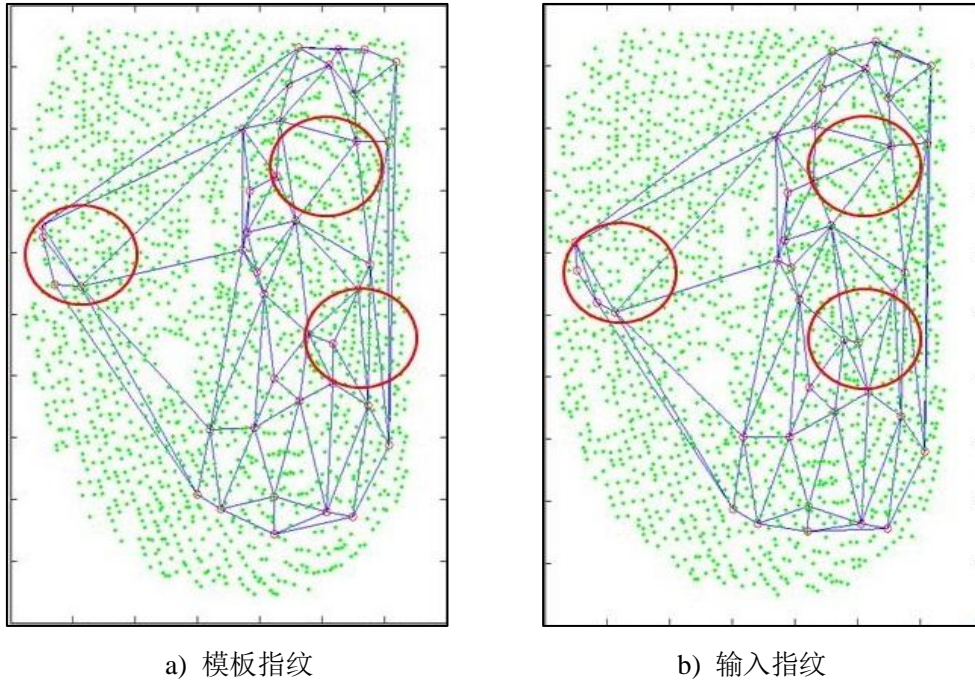


图 3-1 对指纹图像进行三角剖分

通过对比三角剖分的结果，我们找到三角剖分结果一样的三角形，既对比三角形的三个顶点是否相同，然后选择这些三角形内的汗孔进行下一步的匹配对比。对于三角剖分不同区域的汗孔，图中红色圆圈标记的区域，我们认为其区域的形变较大，不予对比。

3.3.2.2 区域配准

配准是匹配中关键的一步，配准效果的好坏，直接影响特征点能否正确匹配到对应的点上。本章中采用分块配准的思想^[34]，对每个块分别进行配准，这样做可以大大减小区域形变对最终匹配结果的影响。

在上一步分块结束后，我们得到了若干对应三角形及三角形三个顶点（三个细节点）的坐标。根据三角形的三个顶点，我们就可以利用其坐标进行配准参数的估计。由于已经采取了分块配准的方式，所以本章选择刚性的四参数坐标转换作为配准所需的转换公式。通常二维直角坐标变换如公式（3-5）所示：

$$\begin{bmatrix} X_1 \\ Y_1 \end{bmatrix} = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} + K \begin{bmatrix} \cos \varepsilon & -\sin \varepsilon \\ \sin \varepsilon & \cos \varepsilon \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \end{bmatrix} \quad (3-5)$$

式中 (X_0, Y_0) ——输入点的坐标;

(X_1, Y_1) ——对应点的坐标;

$(\Delta x, \Delta y)$ ——坐标平移参数;

ε ——偏转角度;

K ——尺度参数。

我们用 p 表示 $K \times \cos \varepsilon$ ，用 q 表示 $K \times \sin \varepsilon$ ，则上式变为公式 (3-6) 所示：

$$\begin{cases} X_1 = \Delta x + pX_0 - qY_0 \\ Y_1 = \Delta y + qX_0 + pY_0 \end{cases} \quad (3-6)$$

这样 $(\Delta x, \Delta y, p, q)$ 就是坐标转换所需的四个参数。根据估计出的 $(\Delta x, \Delta y, p, q)$ ，我们就可以计算每个点转换后的对应点。

参数估计本章采用的是最小二乘法原理^[35-37]。在利用匹配细节点对三角剖分后，得到若干三角形及其顶点坐标。对于每个三角形有三个点，这样就可以得到 6 个方程，我们将其用矩阵的形式表示如公式 (3-7)、(3-8)、(3-9)、(3-10) 所示：

$$B_m = \begin{bmatrix} 1 & 0 & x_m & -y_m \\ 0 & 1 & y_m & x_m \end{bmatrix} \quad (3-7)$$

$$B = [B_1 \quad B_2 \quad B_3]^T \quad (3-8)$$

$$l_n = \begin{bmatrix} X_n \\ Y_n \end{bmatrix} \quad (3-9)$$

$$L = [l_1 \quad l_2 \quad l_3]^T \quad (3-10)$$

式中 (x_m, y_m) ——输入图像细节点的坐标;

B —— 6×4 的矩阵;

(x_n, y_n) ——模板图像对应细节点的坐标;

L —— 6×1 的矩阵。

然后根据最小二乘法原理，采用公式 (3-11) 计算配准参数，

$$Y = \langle B^T B \rangle^{-1} B^T L \quad (3-11)$$

这样我们就得到了该三角形区域的转换参数。对于每个三角形区域，我们都采用这种方式进行参数估计。如图 3-2 所示，然后再利用估计出的参数，对三角形内的汗孔进行坐标转换，接下来就可以进行寻找匹配汗孔点的步骤。

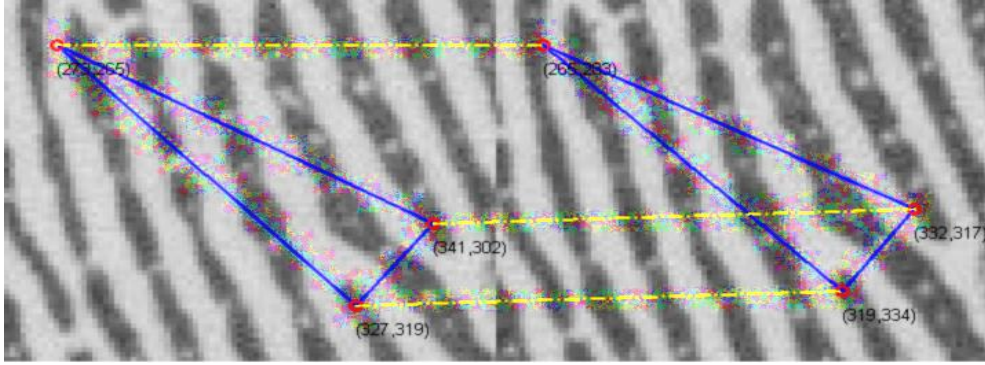


图 3-2 根据对应三角形的顶点坐标估计配准参数

3.3.2.3 基于距离的汗孔匹配

在得到了三角形的配置参数后，我们找到对应三角形内的汗孔。其中优先选择距离细节点较近的汗孔，本章中选择距离细节点较近的 50 个汗孔点。这样选择的原因是距离细节点较近的汗孔点具有较高的区分度。然后根据配准参数对汗孔点集的坐标进行转换，得到了配准后的汗孔点后，我们采用动态规划法进行汗孔点的匹配。匹配的原则是采用基于距离的原则，既如果配准后的两个点的直线距离小于阈值 TD ，则认为这两个点匹配。最后的匹配汗孔数为每个三角形内匹配的汗孔数的总和 M_p 。然后分别统计两个指纹图像中，对应三角形内的汗孔总和 N_1, N_2 。

则这两个指纹图像的匹配得分由公式 (3-12) 求得：

$$S = \frac{2 \times M_p}{N_1 + N_2} \quad (3-12)$$

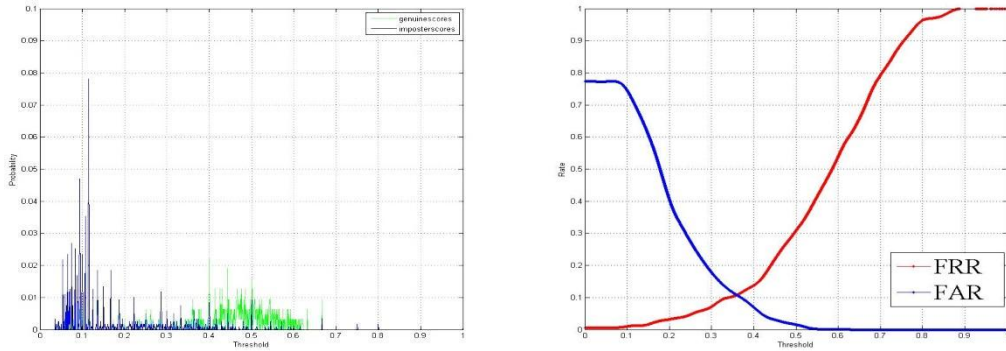
式中 M_p ——对应三角形内的匹配汗孔对数；

N_1 ——输入指纹对应三角形内的汗孔个数；

N_2 ——模板指纹对应三角形内的汗孔个数。

匹配分数主要决定于对应三角形区域内的汗孔匹配效果，匹配得分取的是匹配的汗孔个数除以两个三角形内汗孔个数的平均值。这样可以有效减少个别三角形的误匹配造成的误差，一定程度上提高了的算法的鲁棒性。

使用本章的汗孔匹配算法，我们在一个具有 35 个指头的数据库中进行了测试。为了测试拒识率 (FRR)，我们做了 315 次实验；为了测试误识率，我们做了 1190 次实验，汗孔的得分分布和在不同阈值下的 FRR 和 FAR 曲线如图 3-3 所示。在阈值取 0.32 时，得到的 EER 为 24.902%。



a) 汗孔的得分分布

b) FRR 与 FAR 曲线

图 3-3 汗孔匹配实验结果

3.3.3 算法的时间复杂度

相比目前的汗孔匹配算法，本章提出的基于分块的汗孔匹配算法时间复杂度较低。首先分析 Zhao 等人提出的直接汗孔匹配算法，其粗匹配过程中，要对每一对汗孔进行计算，时间复杂度较高为 $O(n^2)$ 。而采用先配准后匹配的算法，如克日什科夫的算法，在每个配准阶段都要计算所有汗孔点的匹配结果，时间复杂度同样为 $O(n^2)$ 。可以看到时间复杂度较高，已经成为汗孔匹配算法提高的瓶颈。本章提出采用细节节点分块的方法，对汗孔图像进行三角剖分分块，使用每个三角形的顶点进行配准参数的估计，可以大大降低配准阶段的时间复杂度，从而降低汗孔匹配算法的复杂度。分析高分辨率指纹的特点，汗孔的数量庞大，一般一个指纹图像的汗孔数量可以达到 1000 以上，如果采用全部汗孔进行配准，时间复杂度较高，达到 $O(n^2)$ ，另外不能解决局部非线性形变的问题。而细节节点个数相对较少，特别是最终匹配的细节节点对，一般只有几十个，使用这些参数进行配准，时间复杂度为 $O(m)$ ，其中 m 远小于 n ，所以本章提出的算法可以有效减少配准阶段的时间复杂度。

3.4 多特征的融合

由于本身难以克服的缺点和环境干扰等问题，在生物特征匹配时，使用单一特征往往达不到理想的效果，事实证明利用特征融合技术能够有效解决这个问题。本章根据高分辨率指纹的特点，进行指纹方向场，指纹细节节点，指纹脊线，指纹汗孔等的特征融合，以达到最终提高匹配准确率的目的。

目前多生物特征融合主要从采集层、特征层、量化层和决策层四个层面进行

融合，目的是克服单一特征的局限性，使最终匹配效果更好。采集层和特征层融合在采集和提取时就开始融合进而产生新的特征，但是存在兼容性的问题，因而使用并不普遍。匹配级融合通过不同的策略将不同特征的得分进行融合，计算出新的总得分。在现实应用中，匹配级融合具有容易实现，鲁棒性强等优点，因而应用最广泛。决策层融合也具有易于实现的优点，这种方法根据不同特征计算相应的结果进行融合得到最终结果，常见的融合规则有 AND，OR，决策树等。决策层融合的缺点是要求融合的结果为整数，比如两类特征的融合结果都为 0 或 1。

本章使用匹配级融合的方法，在基于匹配值融合的方法中，使用最多也是最简单的是 OS-Rule^[38-40]，包括加法法则（Sum-Rule），乘法法则（Product-Rule），最小法则（Min-Rule），最大法则（Max-Rule），加权平均法则（Weighted Sum-Rule）等。这四种方法思想成熟，易于实现。很多文献从理论和实验证明了这四种方法的优点。例如 Duin^[41]就从理论上证明了在单个特征噪声较大时，加法法则优于乘法法则，在单个特征匹配准确率较高时，乘法法则优于加法法则。本章采用加权平均法则，分别计算基于方向场，基于细节点和基于汗孔的得分，然后采用加权平均的方法计算最终的匹配得分。第一步首先计算方向场匹配得分 S_1 ，细节点匹配得分 S_2 ，和汗孔的匹配得分 S_3 ，对于得到的匹配得分我们采用如下的线性归一化公式，见公式（3-13）：

$$S_{normalized} = \frac{S - \min(S)}{\max(S)} \quad (3-13)$$

最后每种特征的匹配得分被归一化到一个范围内。由于每种特征相对独立，且用独立的算法匹配，所以在融合后能够很好地融合每种特征的优点，进而提高最终的识别准确率。加权融合的方法^[42]采用公式（3-14）求得：

$$S = \alpha \times S_1 + \beta \times S_2 + (1 - \alpha - \beta) \times S_3 \quad (3-14)$$

计算最终的匹配得分。结果表明，相比单独使用方向场，细节点，汗孔进行匹配，使用方向场，细节点，汗孔结合的匹配结果，可以有效提高匹配的准确率。融合后的匹配得分有效降低了个别特征的受特定环境影响的缺陷，使算法的准确率和鲁棒性均有所提高。

为了测试本章提出的匹配算法，在一个较小的指纹库上做了初步实验，该库包含 35 个指头，每个指头分两个时期，每个时期 3 张图像，共 210 个指纹图像，采集区域为指尖 6.5×4.9 毫米区域。为了测试拒识率（FRR），我们设计了 Genuine 集为每个指头第一时期采的指纹分别与第二时期的指纹图像两两匹配，共 35×9=315 次匹配。为了测试误识率（FAR），我们设计 Impostor 集为每个指头的第一

时期的第一张图像与其他指头的第二时期的第一张图像匹配，共 $35 \times 34 = 1190$ 次匹配。基于以上数据库，我们做了如下实验，单独使用方向场的匹配实验，单独基于细节节点的匹配实验，本章提出的汗孔匹配方法实验和本章提出的融合三种特征的匹配实验。表 3-1 显示了四个实验的结果，相应识别准确率分别达到了 67.6%，74.51%，75.10%，78.85%。

表 3-1 融合效果的对比

算法	正确识别个数	EER
仅使用方向场匹配	1017	32.40%
仅使用细节节点匹配	1121	25.49%
基于分块的汗孔匹配	1130	24.90%
融合三种特征匹配	1187	21.15%

本章采用加权平均的方式进行融合，采用公式 3-10。因为细节节点和汗孔的匹配效果较好，所以在权值的选择时，细节节点和汗孔的权值较大。根据我们得到的匹配数据实验得到结果如图 3-4 所示，在细节节点权值在 (0.5, 0.7) 之间，汗孔权值在 (0.26, 0.5) 之间时，等误率 (EER) 最低。最终本章融合权值选择为 $\alpha = 0.34$ ， $\beta = 0.62$ ，对应得到的 EER 为 21.15%。

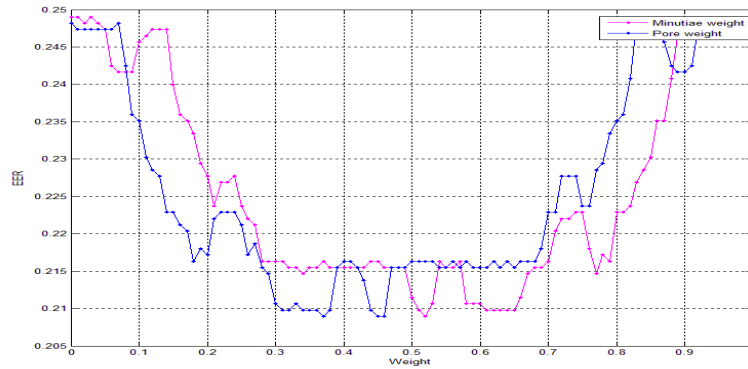
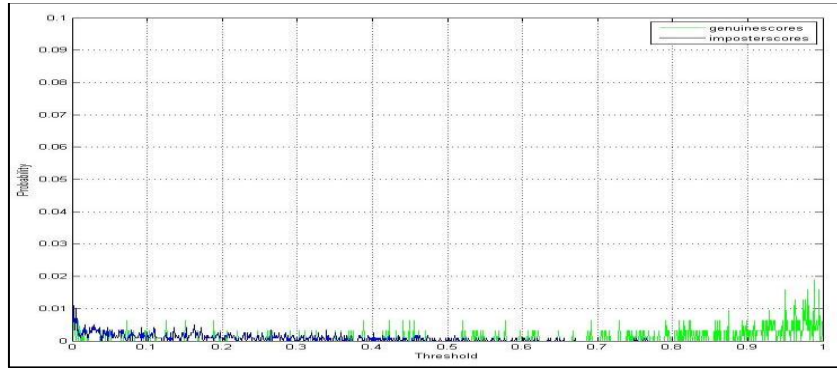
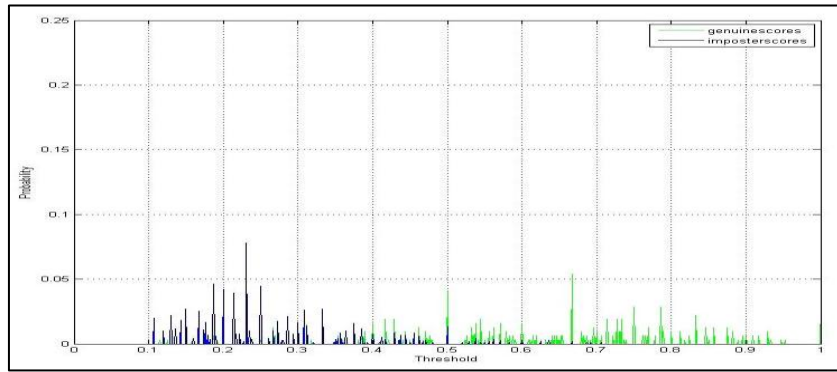


图 3-4 对应不同权值选择时的 EER 走势

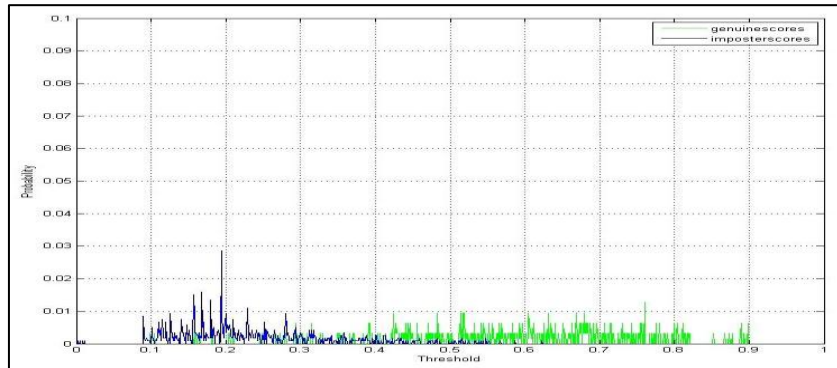
为了对比分类效果，我们在图 3-5 中统计了基于方向场匹配，基于细节节点匹配，融合三种特征匹配的得分分布情况。对比三种方法，基于方向场的得分 Genuine 集的得分和 Impostor 集的得分重合较多，说明分类效果不明显，基于细节节点的匹配得分，Genuine 集和 Impostor 集得分分布比较分散，但仍有一部分的 Impostor 集的得分与 Genuine 集得分重合，这部分得分说明分类的效果仍有待加强。分析本章提出的融合三种特征匹配的得分分布，Genuine 集合 Impostor 集的得分重合部分较少，说明较好的实现了分类。



a) 方向场的得分分布



b) 细节节点的得分分布



c) 融合后的得分分布

图 3-5 得分分布对比

从以上初步结果可以得到，我们提出的基于细节节点分块后融合的方法，相比单独使用细节节点或方向场的匹配方法识别率更高。

3.5 本章小结

本章提出了一种基于细节节点三角剖分分块进行汗孔匹配，并采用多特征融合的指纹匹配算法。算法主要根据细节节点进行指纹的分块，并采取区域配准，区域匹配的方法计算汗孔的匹配得分，最后根据三种特征的匹配得分进行加权平均得

到最后的匹配得分。本章通过实验证明了本算法采用分块配准的方式，很好地解决了局部非线性形变对匹配结果的影响。相比同类的算法准确率有所提高。然而本章中的算法也有很多需要改进的地方。其中包括：

- （1）指纹分块依赖于细节节点的匹配结果，对于细节节点匹配点较少的情况下，汗孔匹配结果受影响较大。
- （2）算法对于图像质量较差的指纹处理能力尚需改进。

第 4 章 基于脊线的汗孔匹配

4.1 引言

汗孔由于信息量大如图 4-1 所示，便于提取等特征，成为目前高分辨率指纹识别的主要研究方向。本章结合目前高分辨率指纹识别的研究现状，提出了一种基于脊线的汗孔匹配算法。

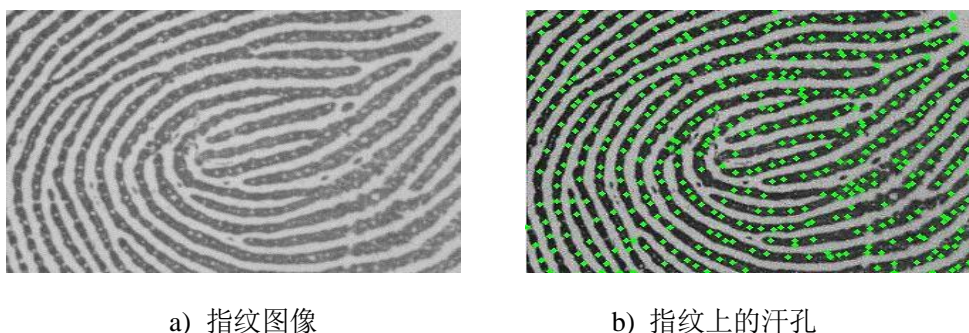


图 4-1 指纹上的汗孔信息

分析目前的指纹匹配算法，指纹的匹配主要面临由于手指较干或较湿造成的指纹图像质量较低，采集时造成的形变，扭曲等问题。而高分辨率指纹的匹配也面临着自己的问题，目前高分辨率指纹的匹配用到的主要是汗孔。如图 4-2 所示，由于高分辨的指纹图像，使原来的局部形变也被相应放大，这对于汗孔这样的特征点影响较大，由于汗孔在不同时刻的可能张开也可能闭合，所以汗孔在每次采集的一致性不好，这也给汗孔的匹配造成了一定的麻烦，汗孔之间很可能造成误匹配。结合这些问题，本章提出一种基于脊线的汗孔匹配算法，将脊线的匹配和汗孔的匹配结合在一起，使用匹配的脊线对汗孔进行坐标校准，然后与细节点和方向场的特征进行融合，得到了较好的效果。

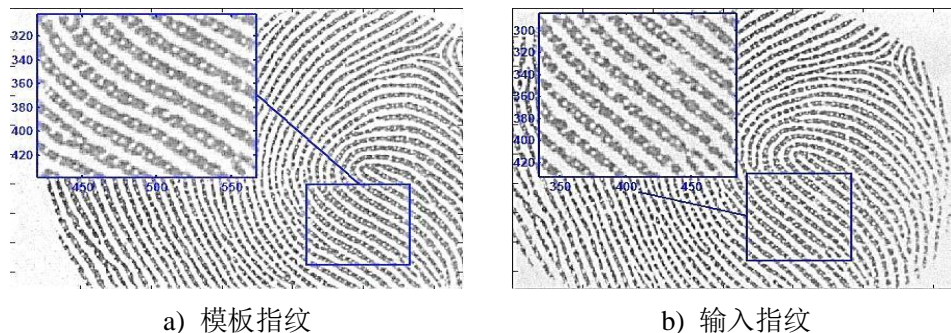


图 4-2 指纹的局部形变

本章中结合冯建江等人提出的脊线匹配算法^[43]，提出一种结合细节点的快速脊线匹配算法进行脊线的匹配。目前对脊线的研究较多^[44,45]，但主要都作为配准的根据^[46,47]，用以解决指纹的非线性形变。目前还没有用于结合汗孔的指纹匹配。本章提出的算法结合脊线匹配，汗孔匹配得到最终的结果作为汗孔的匹配得分。算法的第一步是前期处理，提取脊线特征，脊线和汗孔的相对关系；第二步，脊线的匹配；第三步，根据匹配脊线的结果，按脊线进行脊线上的汗孔的匹配。在前期处理阶段，根据细化后的指纹，我们按照一定的间距对脊线进行采样，最终脊线被保存为一串采样点的集合。在脊线采样的同时我们把脊线间的相互关系一起提取出来。在脊线匹配阶段，我们使用结合细节点的快速匹配算法找到匹配的脊线，并根据匹配的脊线估计每对脊线的配准参数。最后在汗孔的匹配阶段，我们根据脊线的匹配结果，根据每条脊线的配准参数，对每条脊线上的汗孔进行配准，再根据距离的原则进行匹配。本算法中汗孔的匹配使用的是根据每条脊线的匹配点重新估计的配准参数，相比目前大多数指纹匹配算法中使用的整体匹配，按照每条脊线分别进行坐标校准，是采用局部的配准的思想，能够更好地解决非线性形变对汗孔匹配的影响。

在本算法中，最终的汗孔匹配得分，要结合相应脊线的匹配程度，汗孔的匹配得分为匹配脊线上的汗孔得分乘以脊线的匹配得分。

4.2 前期处理

4.2.1 特征的提取

本章中用到的特征包括细节点，脊线，汗孔。其中脊线的提取过程采用冯建江等人提出的提取算法。

细节点和脊线的提取：采用传统的指纹细节点提取算法，两个部分同时完成。对于每一个指纹图像：第一步，对指纹图像进行归一化；第二步，找到指纹的 ROI 区域，进行图像分割；第三步，指纹的方向场的计算；第四步，根据方向场，对指纹图像进行增强；第五步，指纹图像的二值化和细化；第六步，根据细化后的图像提取细节点和脊线信息，脊线提取的效果如图 4-3 所示。细节点保存为 $((x_1, y_1, \theta_1)^T, \dots, (x_m, y_m, \theta_m)^T)$ 的形式，其中 x, y 为细节点的坐标信息； θ 为细节点的方向。脊线提取根据一定间距提取相应的脊线采样点坐标，则一条脊线提取后保存为 $((x_1, y_1)^T, \dots, (x_n, y_n)^T)$ 的形式，其中 (x_i, y_i) 为采样点的坐标， n 为采样点的个数。

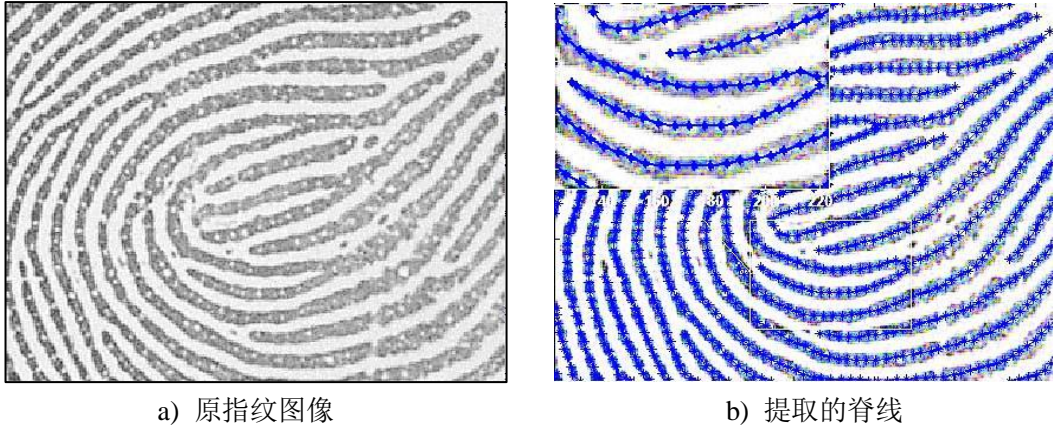


图 4-3 脊线的提取

汗孔的提取采用分块自适应的汗孔提取方法。汗孔的提取步骤和细节点类似。进行图像分割，选取指纹所在区域；计算指纹图像的方向场；构造两个高斯滤波器；分别对指纹图像进行两次高斯滤波，得到增强的汗孔和脊线；然后根据增强结果去掉伪汗孔得到提取的汗孔信息。提取的汗孔特征为每个汗孔的坐标。然后根据提取出来的脊线，按照距离的远近对汗孔进行分类，找到汗孔所属的脊线。最终汗孔保存如公式（4-1）所示：

$$P = \left((x_1, y_1, n_1)^T, \dots, (x_k, y_k, n_k)^T \right) \quad (4-1)$$

式中 (x_i, y_i) ——汗孔的坐标；

n_i ——汗孔所属脊线号。

4.2.2 脊线的表示

由于图形质量和提取算法的缺陷的影响，最终提取的脊线会出现很多复杂的结构，比如环，桥。为了最终脊线匹配的方便，本章采取将脊线拆分的方式，将复杂的脊线拆分和清理：首先对于闭合的环，任意选择一点，将其断开，使其成为一条脊线；其次对于具有分叉点的脊线，在分叉点处将其分成三条脊线；最后对于长度过短的脊线，我们将其删除。

在清理后，对每条细化后的脊线进行采样提取。采样的方法是根据一定间隔每隔一段采集一个脊线点坐标，最终获得一串采样点的坐标值。采样的脊线保存如公式（4-2）求得：

$$SP = \left(\left(X_{m,n}, Y_{m,n} \right)_{m=1}^Q \right)_{n=1}^P \quad (4-2)$$

式中 (x_i, y_i) ——采样点的坐标；

M ——脊线条数；

N ——第 m 条脊线的采样点数。

4.2.3 脊线之间的关系

本章中的脊线匹配算法用到了相邻脊线的关系，其表示了脊线间的相邻关系。本章中脊线已经被采样成为一串坐标点，脊线间的相邻关系采用脊线间采样点的关系来表示脊线间的关系。

脊线间关系的确定，是扫描脊线的每个采样点，根据该点的脊线方向画两条长度为 30 个像素的垂线段。在脊线的每个方向上，第一条与垂线相交的脊线，就是这条脊线在这个点的相邻脊线。另外，相邻脊线被分为左脊线和右脊线，其中左脊线定义为当前脊线左侧的相邻脊线，右脊线定义为当前脊线的右侧的脊线。脊线的相邻关系如图 4-4 所示。

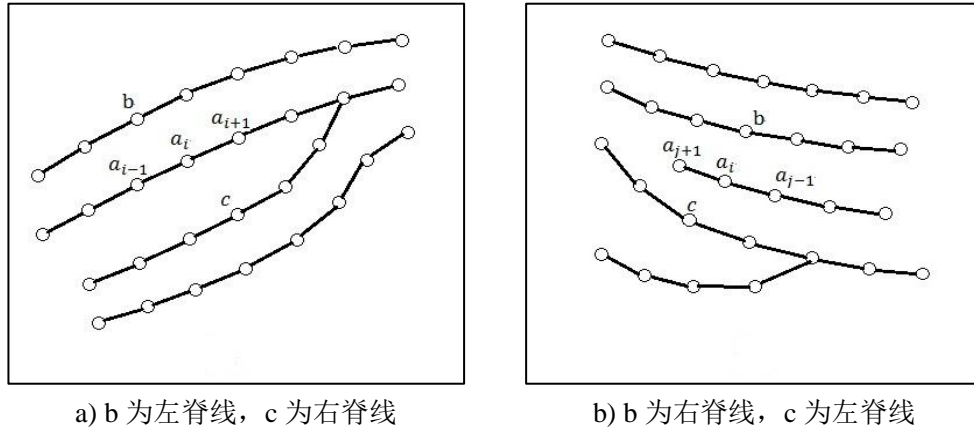


图 4-4 两种脊线的相邻情况

脊线的相邻关系最终保存为向量形式。最终脊线采样点的提取信息由 $[x, y]$ 扩充为 $[x, y, lr, rr]$ 其中 x, y 为采样点的坐标， lr 为该采样点对应的左脊线号， rr 为该采样点对应的右脊线号。如果一个采样点没有对应的相邻脊线，则把 lr, rr 都设置为 -1。

4.3 脊线的匹配

本章中脊线的匹配采取基于细节点的快速脊线匹配算法。首先根据脊线的相似度找到相似的脊线，然后对这些相似脊线进行配准和匹配。由于细节点匹配相对准确而快速，本章脊线的配准，采用细节点的配准结果。配准后，采用动态规划法找到距离相近的采样点，最后根据采样点的匹配个数决定两条脊线的匹配的得分。

4.3.1 脊线相似度

在脊线匹配的第一步，首先找到具有一定相似度的脊线。这样我们可以找到可能匹配的脊线。对于每一对相应脊线 r_1 , r_2 ，我们比较取样后的点的坐标。假设 s_1 , s_2 为相应脊线的起始点， e_1 , e_2 为相应脊线的终结点。首先我们根据脊线的采样点个数，判断脊线的长短。假设 r_1 是较短的那条脊线，我们根据 r_1 脊线的终结点 e_1 找到其在脊线 r_2 上的对应投影点 p_1 。两条脊线的相似度由公式 (4-3) 求得：

$$simi = 1 - |d_1 - d_2| / \max(d_1, d_2) \quad (4-3)$$

式中 d_1 —— r_1 的起始点到终结点的欧式距离；

d_2 —— r_2 的起始点到投影点 p_1 的欧式距离。

4.3.2 脊线的重新配准估计

在得到匹配的脊线后，我们根据匹配的采样点重新估计脊线间的转换参数。由于脊线是一条条的线段，所以本章采用四参数变换作为转换公式。四参数变换具有计算简单，速度快的优点，具体参照第三章中介绍的方法。每次找到匹配的脊线，就得到了两个匹配的点集 B_1 , B_2 ，根据最小二乘法原理，由公式 (4-4) 得：

$$T = \left(B_1^T B_1 \right)^{-1} B_1^T B_2 \quad (4-4)$$

我们就得到了新的配准参数。这样我们就可以根据新的配准参数进行脊线上的汗孔的配准。因为对每条脊线都进行配准的重新估计，这样估计出来的配准参数具有更准确的结果。

4.3.3 匹配

得到了提取的细节点和脊线后，我们开始脊线的匹配。首先建立一个队列 *RidgeList* 保存匹配的脊线号，重新估计的配准参数和匹配的采样点对。再建立一个队列 *CRQueue* 用于存放可能匹配的脊线对，算法不断生成候选脊线，并存入 *CRQueue* 中，然后不断取出 *CRQueue* 中的候选脊线进行匹配，直到 *CRQueue* 中为空，最后根据 *RidgeList* 中的候选脊线计算脊线匹配的结果。算法的流程如下：

- (1) 根据匹配细节点估计配准矩阵 T ；
- (2) 扫描所有脊线对，计算相似度，找到可能匹配的候选脊线对 (r_1, r_2) ，将 (r_1, r_2, T) 存入队列 *CRQueue* 中；
- (3) 如果队列 *CRQueue* 为空，转到(4)；否则， (r_1, r_2, T) 出队；
- (4) 根据配准矩阵 T_i 对脊线 $r_{1,i}$ 和脊线 $r_{2,i}$ 进行配准，使用动态规划法对脊线 $r_{1,i}$

和脊线 $r_{2,i}$ 进行匹配计算。如果没有脊线匹配，转到 (3)；

(5) 对匹配的脊线，根据匹配点进行配准参数的重新估计，并保存到 *RidgeList* 中；

(6) 扫描匹配脊线的左右邻脊线，生成新的候选脊线，并入队到 *CRQueue* 中；

(7) 转到 (3)；

(8) 根据 *RidgeList* 中的结果计算脊线的匹配得分。

以上为算法的大体流程，最终的结果保存在 *RidgeList* 中，其中包含了匹配脊线对，重新估计的配准参数，和对应匹配点。

4.3.3.1 寻找可能匹配脊线

生成候选脊线主要有两种情况，一种是在匹配初期扫描所有的脊线查找候选脊线；另一种是扫描匹配脊线的左右邻接脊线，生成新的候选脊线。

在匹配初期，用每条输入指纹的脊线与每一条模板指纹的脊线进行比较。比较的根据是计算这两对脊线的相似度，我们根据公式 4-3 的方法计算两条脊线的相似度。在匹配初期，我们设定当两条脊线的相似度 $similarity > 0.9$ 时，则认为这两条脊线可能匹配。为了减少不必要的计算，我们又添加了额外的限制条件，目的是找到尽可能相似的脊线。当两条脊线的长度相差超过 1: 1.2 时，我们认为这两条脊线不相似。当脊线的长度小于 10 时，我们不予比较这两条脊线，对应的汗孔数量较少。在这两条限制条件下，我们找到尽可能相似的脊线。严格的限制条件会使一部分脊线被排除在外，这些脊线会在接下来扫描匹配脊线邻接脊线的步骤重新找到。

在找到匹配脊线后，我们要扫描匹配脊线对的左右邻接脊线，并根据其相似度，长度进行比较。在这部分，由于匹配脊线邻接脊线的匹配的可能性较高，我们适当放宽对其相似度的限制，我们认为 $similarity > 0.8$ ，长度相差不超过 1: 1.5 的脊线都是相似脊线。在找到相似脊线后，我们把这些可能匹配的脊线放入 *CRQueue* 中，其配准参数为根据匹配脊线的点重新计算估计得来，也一并放入 *CRQueue* 中。

4.3.3.2 动态规划法脊线匹配

寻找脊线的匹配部分的任务是找到两条脊线的最大匹配采样点数如图 4-5 所示。本章采用动态规划法查找两条脊线的匹配点。由于我们已经得到可能匹配脊线的配准参数，在配准之后，我们只要根据距离的原则就能得到匹配的采样点。我们认为当两个配准后的点距离小于一个阈值 dis 时，就认为这两个点匹配。查找匹配点的流程如下：

(1) 从 $CRQueue$ 中取出一个脊线对 $\{a_i\}_{i=1}^M, \{b_j\}_j^N$ 其中 a_i, b_j 为对应脊线的采样点, 和配准参数 T 。根据配准参数 T , 我们对两条脊线进行配准;

(2) 建立二维矩阵 $EuDis(i, j), i=1, \dots, M; j=1, \dots, N$, i 为对应模板脊线的采样点个数, j 为对应输入脊线的采样点个数, 如表 4-1 所示。对应矩阵点 $EuDis(i, j)$ 的值表示采样点 a_i, b_j 之间的距离关系, 当距离小于阈值 dis 时, $EuDis(i, j)=1$, 当阈值大于 dis 时, $EuDis(i, j)=0$;

(3) 接下来使用动态规划法找到两条脊线的匹配采样点个数。既查找矩阵 $EuDis$ 中的最长的为 1 的点的个数。当匹配点满足采样点序号单调上升时, 得到相应的匹配采样点;

(4) 比较匹配采样点的个数, 如果匹配采样点的个数小于两条脊线重合部分的 80%, 则认为这两条脊线没有匹配。

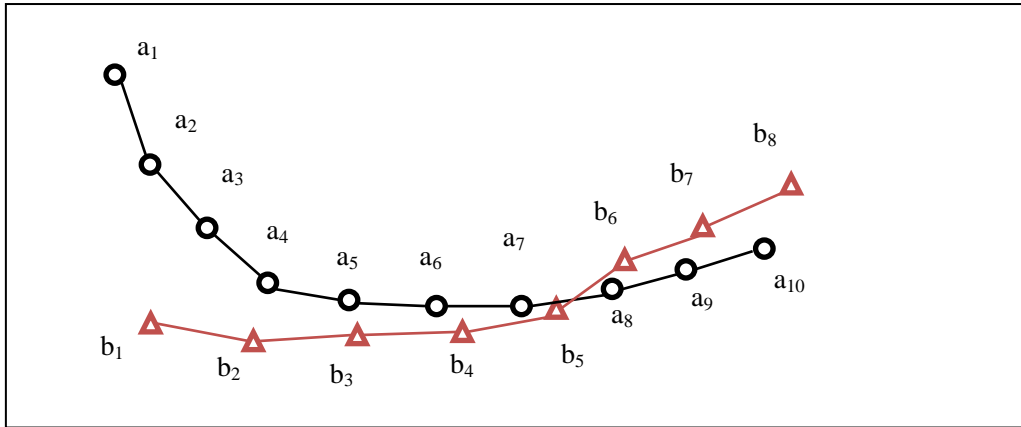


图 4-5 两条脊线及脊线上的采样点

表 4-1 根据对应点距离计算距离矩阵

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
b_1	0	0	0	0	0	0	0	0	0	0
b_2	0	0	0	0	0	0	0	0	0	0
b_3	0	0	0	0	1	0	0	0	0	0
b_4	0	0	0	0	0	1	1	0	0	0
b_5	0	0	0	0	0	0	1	1	0	0
b_6	0	0	0	0	0	0	0	1	0	0
b_7	0	0	0	0	0	0	0	0	1	0
b_8	0	0	0	0	0	0	0	0	0	0

4.3.3.3 计算匹配得分

根据以上的步骤，我们就得到了匹配脊线对列 *RidgeList*，如图 4-6 所示，其中保存了每一对匹配的脊线号，重新估计的配准参数，及匹配的采样点对应矩阵。根据匹配脊线队列，我们用匹配脊线个数计算最终匹配得分。匹配得分计算如公式 (4-5) 所示：

$$RS = 2m_r / (T_r + Q_r) \quad (4-5)$$

式中 m_r ——匹配的脊线条数；

T_r ——模板指纹的脊线条数；

Q_r ——输入指纹的脊线条数。

最终的脊线匹配得分为匹配的脊线条数除以模板指纹和输入指纹的脊线条数和的一半。

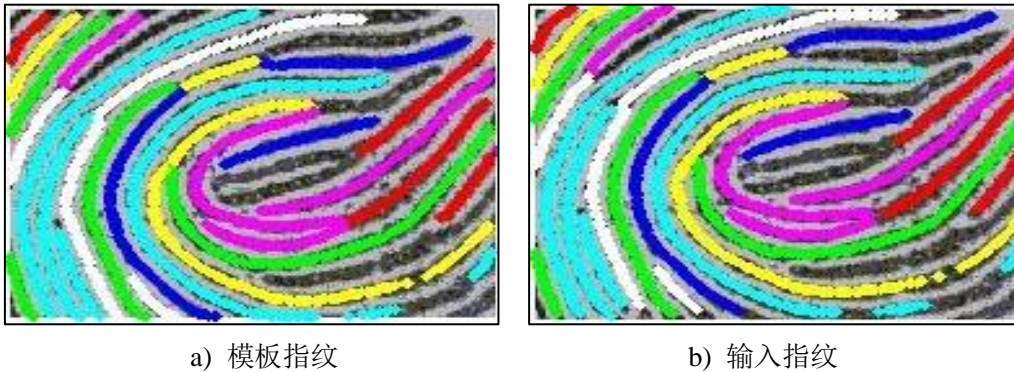


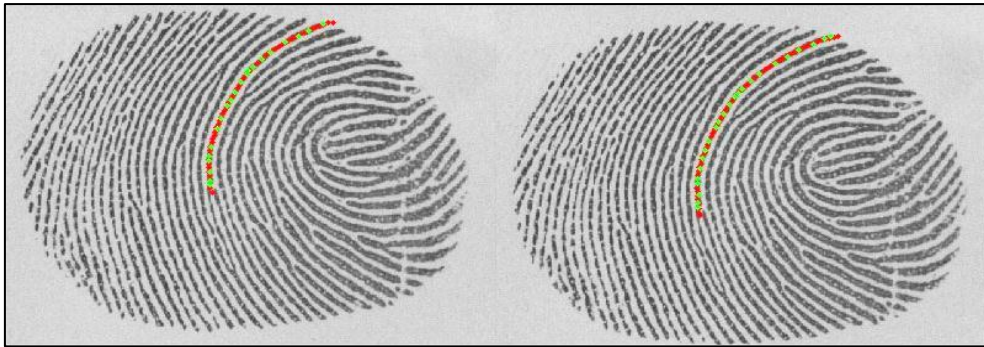
图 4-6 脊线的匹配

4.4 基于脊线的汗孔匹配

在得到了脊线匹配的结果后，我们根据脊线进行汗孔的匹配。如图 4-7 所示，匹配的汗孔事先用已经提取好的脊线进行了分类，既获得了每个汗孔的所属脊线。再根据匹配脊线的重新估计的配准参数对汗孔进行配准。配准后的汗孔，只需要根据距离的原则查找相近的汗孔作为匹配汗孔。算法的流程如下：

- (1) 从匹配的脊线对列中取出一对匹配脊线及其重新估计后的配准参数；
 - (2) 根据匹配脊线号，找到相应脊线上的汗孔；
 - (3) 使用该对脊线上的配准参数对汗孔进行重新配准；
 - (4) 使用动态规划法对配准后的汗孔进行匹配计算；
 - (5) 统计匹配脊线上的汗孔的匹配汗孔数，计算每条脊线的匹配汗孔数；
 - (6) 最终的汗孔匹配得分为脊线的匹配得分乘以匹配脊线上的汗孔匹配得分。
- 本算法的最终匹配得分是根据匹配脊线得分和相应脊线上的汗孔得分相乘得

到的，是结合了脊线和汗孔两种指纹信息的匹配结果。相比传统的汗孔匹配得分，基于脊线的汗孔匹配使用每对匹配脊线重新估计脊线上的配准参数，根据匹配脊线重新估计的配准参数，采用了局部配准的方法，比采用整体配准的方法能够更好地解决非线性形变带来的影响。而由于最终匹配的是匹配脊线上的汗孔，汗孔的数量大大减少，相比直接汗孔匹配算法，计算复杂度也大大降低。经过香港理工大学的高分辨率指纹库测试，本章的算法具有较高的准确率和较低的复杂度。



a) 找到匹配脊线和相应汗孔



b) 按照相应脊线估计配准参数，对汗孔配准



c) 按照距离远近，找到匹配的汗孔

图 4-7 脊线上汗孔的匹配流程

为了测试本章提出的匹配算法，在一个较小的指纹库上做了初步实验，该库包含 35 个指头，每个指头包含 6 个图像，共 210 个指纹图像，采集区域为指尖 6.5

×4.9 毫米区域。基于以上数据库，我们做了如下实验，仅以方向场为特征的匹配实验，仅用细节点为特征的匹配实验，本章提出的汗孔匹配方法实验和本章提出的融合三种特征的匹配实验。在每组试验中，为了测试拒识率（FRR），我们设计了 Genuine 集为每个指头的不同时期的图像两两匹配，共 $9 \times 35 = 315$ 次匹配。为了测试误识率（FAR），我们设计 Impostor 集为每个指头的第一时期第一张图像与其他指头的第二时期第一张图像匹配，共 $35 \times 34 = 1190$ 次匹配。图 4-8 为汗孔匹配结果，显示了在不同阈值下的 FRR 和 FAR，在阈值取 0.017 时，得到的 EER 为 24.566%。

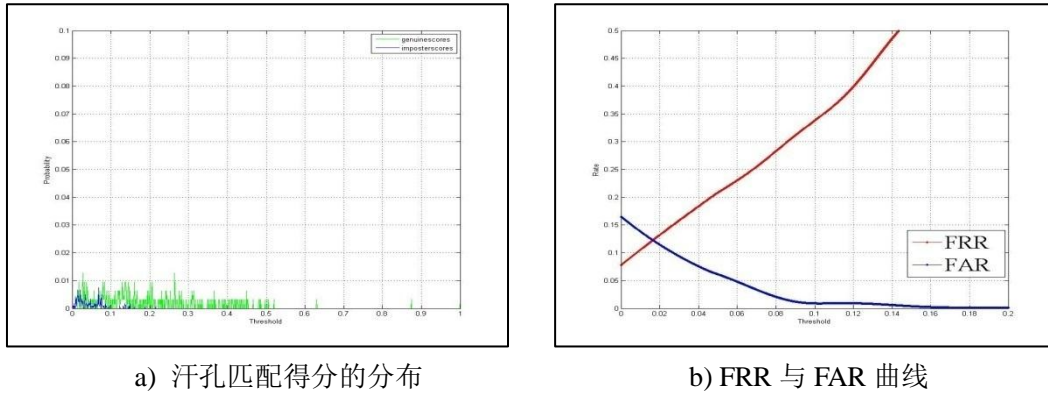


图 4-8 汗孔匹配的实验结果

4.5 多特征的融合

本章使用匹配级融合的方法，分别将方向场，细节点，汗孔的匹配得分融合在一起，然后根据融合公式进行综合计算得到最终的结果。具体的融合方法详见 3.4 节所示。结果表明，相比单独使用方向场，细节点，汗孔进行匹配，使用方向场，细节点，汗孔结合的匹配结果，可以有效提高匹配的准确率。融合后的匹配得分有效降低了个别特征的受特定环境影响的缺陷，使算法的准确率和鲁棒性均有所提高。为了测试融合的效果，我们使用上一节所用的数据进行了实验。

如表 4-2 所示，四个实验的相应识别准确率分别达到了 67.60%，74.51%，75.43%，83.37%。

表 4-2 融合效果对比

算法	正确识别个数	EER
基于方向场的匹配	1017	32.40%
基于细节点的匹配	1121	25.49%
基于脊线的汗孔匹配	1135	24.57%
融合后	1255	16.66%

本章采用第三章介绍的融合策略，在融合时，因为细节点和汗孔的匹配效果较好，所以对细节点和汗孔选择较大的权值。如图 4-9 所示，在汗孔权值取 (0.7, 0.8) 之间，细节点权值选择 (0.18, 0.25) 之间时，等误率最低。最终我们选择 $\alpha = 0.76$ ， $\beta = 0.20$ 作为最终的融合权值。

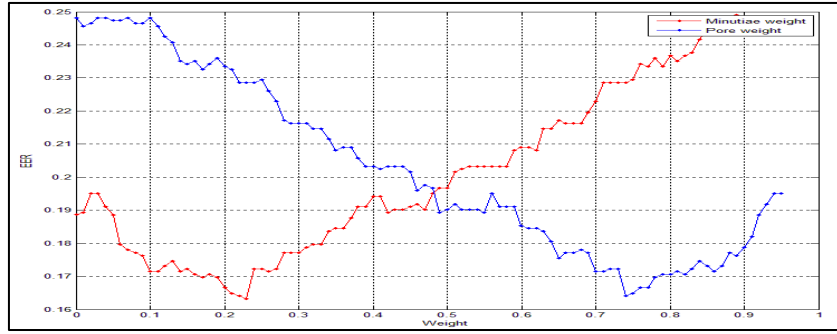
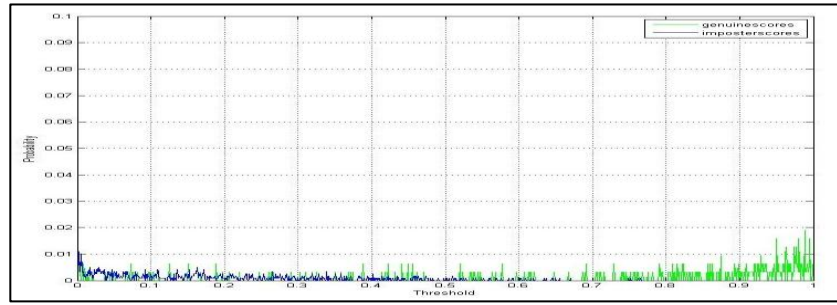
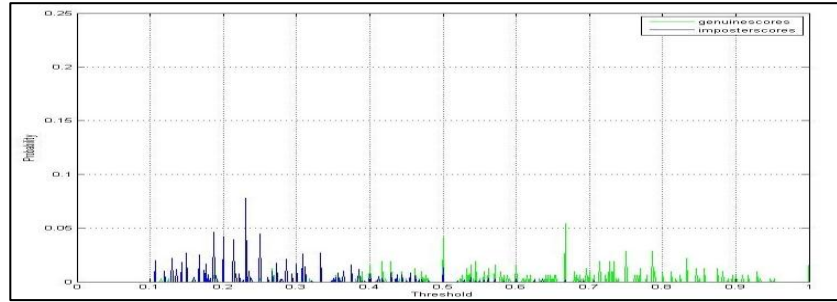


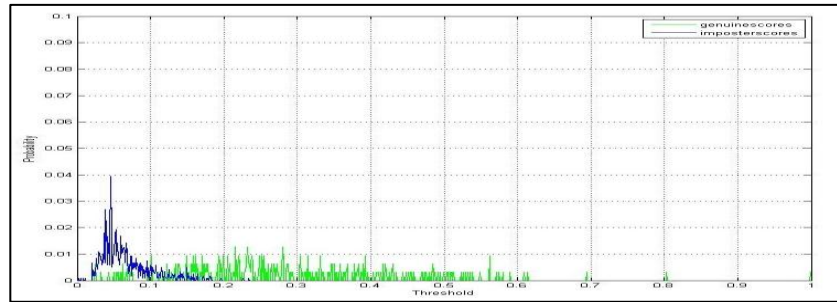
图 4-9 不同权值选择下的等误率



a) 方向场的得分分布



b) 细节点的得分分布



c) 融合三种特征的得分分布

图 4-10 得分分布对比

如图 4-10 统计了基于方向场匹配, 基于细节点匹配, 融合三种特征匹配的得分分布情况。对比三种方法, 基于方向场的得分 Genuine 集的得分和 Impostor 集的得分重合较多, 区分度不高; 基于细节点的匹配得分, 得分分布比较分散, 但 Impostor 集与 Genuine 集得分分布仍有重合。分析融合三种特征匹配的得分分布, Genuine 集合 Impostor 集的得分重合部分较少, 分类效果较好。

分析以上结果, 本章提出的基于脊线的汗孔匹配加融合的方法, 相比单独基于细节点, 方向场的匹配识别准确率有所提高, 并且整体性能优于单独使用一种特征的匹配方法。

4.6 本章小结

本章提出了一种基于脊线的汗孔匹配算法对指纹的汗孔进行匹配, 然后融合细节点, 方向场等信息得到最终的匹配得分, 得到了较好的匹配结果。算法首先根据脊线进行匹配, 找到匹配的脊线, 然后根据匹配的脊线估计出脊线间的配准参数。汗孔匹配阶段, 根据匹配脊线找到脊线上的汗孔, 然后根据重新估计的配准参数对汗孔进行坐标校准。实验证明采用按照脊线估计配准的方式能够更好地对汗孔进行配准。在得到汗孔的配准结果后, 采用动态规划法对汗孔进行匹配。最终的汗孔匹配得分为匹配的脊线分数乘以脊线上的汗孔得分, 可以说这个得分结合了汗孔和脊线两种信息的匹配结果。最后对汗孔匹配得分和细节点, 方向场的匹配结果进行了融合, 融合策略采用匹配级融合。得到的最终结果, 相比同类算法有较低的复杂度和较高的准确率。

然而本算法仍有值得改进之处, 首先, 算法中脊线匹配算法, 对于图像质量差的指纹鲁棒性不好, 当遇到脊线提取效果不好的情况, 会拒绝很多应该匹配的脊线; 其次, 汗孔的匹配依赖于脊线的匹配结果, 受脊线匹配影响较大, 对于误匹配的脊线, 可以采用加权的方式, 降低误匹配的脊线对最终匹配结果的影响。

第 5 章 实验结果与分析

5.1 引言

本文基于分治思想提出了两种汗孔匹配算法，分别为基于细节点分块的汗孔匹配和基于脊线的汗孔匹配，并用程序进行了实现。为了测试本文所提出的两种基于分治思想的汗孔匹配识别算法的性能，我们在香港理工大学的高分辨率指纹数据库上进行了实验，实验包括本文提出的两种汗孔匹配算法与现有类似汗孔匹配算法的性能比较以及使用分块配准和整体配准的效果对比实验，用来验证本文提出的基于分治思想的配准和匹配效果。还比较了单独使用细节点的识别方法和融合多特征集的指纹识别方法，用来验证本文提出的两种基于分治思想的汗孔匹配识别算法的性能。

5.2 实验数据库

本文中所用的数据库为从香港理工大学生物计算中心网站下载的高分辨率指纹库。该指纹库的分辨率为 1200dpi。本数据库包含两个部分，第一部分包含 148 个指头的 1480 个指纹图像。每个指纹分两个时期采集，每个时期每个指头采集 5 张图像，两个时期间隔 2 个星期，每个指头有 10 张图像。148 个指纹就包含了 1480 个指纹图像。第一部分 DBI 的指纹图像大小为 320×240 ，包含的指纹中心区域大小为指尖的 $6.5\text{mm} \times 4.9\text{mm}$ 。指纹库的第二部分为同样方法采集的整个指纹图像 DBII，图像大小为 640×480 。这些图像的汗孔采用改进的汗孔提取算法提取。为了测试本文中提出的基于分块汗孔匹配算法和基于脊线的汗孔匹配算法的性能。我们做了如下性能测试。

5.3 性能测试

5.3.1 多特征融合测试

本文提出的两种算法都是基于多特征融合的方法实现。为了测试融合的效果，我们使用数据库 DBII 中的 1480 个指纹做了如下两组实验。第一组实验，以方向场为特征做匹配，以细节点为特征做匹配，基于分块的汗孔匹配，将三种特征融合后做匹配。在第二组实验中，做了仅用方向场匹配，仅用细节点匹配，基于脊线的汗孔匹配和三种特征融合后的匹配实验。在每组实验中，为了测试拒识率（FFR），对同一手指的不同图像做了 6660 次实验，为了测试误识率（FAR），对

不同手指做了 21756 次实验。图 5-1 和图 5-2 为两组实验的 ROC 曲线对比。两组实验的 EER 对比如表 5-1 和表 5-2 所示。

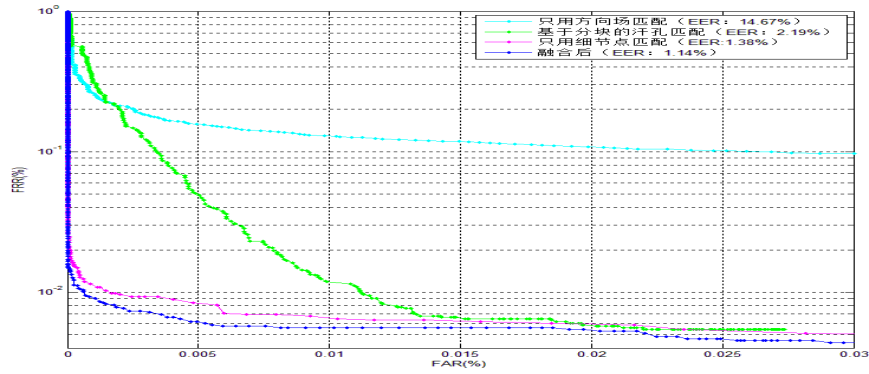


图 5-1 基于分块汗孔匹配融合 ROC 曲线对比

表 5-1 使用方向场、细节点、分块匹配汗孔、融合三种特征后的等误率对比

特征	EER
方向场	14.67%
细节点	1.38%
分块匹配汗孔	2.19%
融合后	1.14%

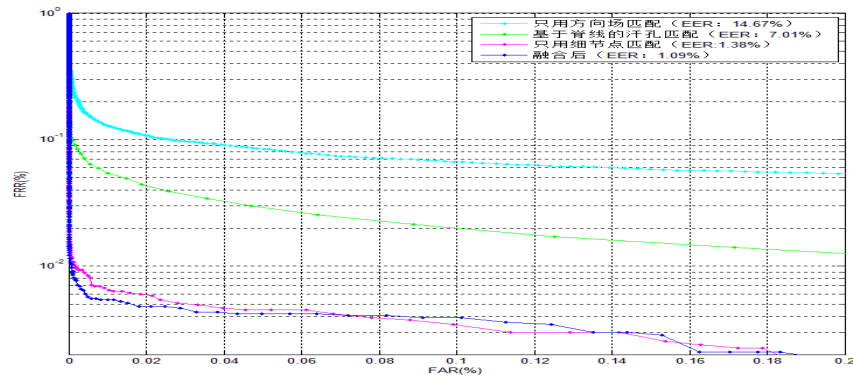


图 5-2 基于脊线汗孔匹配的融合 ROC 曲线对比

表 5-2 使用方向场、细节点、按脊线匹配汗孔、融合三种特征的等误率对比

特征	EER
方向场	14.67%
细节点	1.38%
按脊线匹配汗孔	7.01%
融合后	1.09%

在三种特征的融合阶段，本文使用的是加权得分融合的策略。为了测试融合的效果，我们使用三种特征的匹配结果做了如下两组实验，每组实验分别使用 min rule（最小值原则），max rule（最大值原则），product rule（乘法原则），weighted sum rule（加权融合原则）四种融合策略进行融合。表 5-3 显示了两种算法下分别使用 min rule（最小值原则），max rule（最大值原则），product rule（乘法原则），weighted sum rule（加权融合原则）策略进行融合后的等误率（EER）。实验数据显示，在每组实验中，使用加权融合的方式融合的等误率最低。

表 5-3 采用不同融合策略的等误率（EER）对比

算法（EER）	MIN	MAX	PRODUCT	WEIGHTED SUM
基于细节点分块汗孔匹配后融合	2.44%	4.18%	1.65%	1.14%
基于脊线的汗孔匹配后融合	3.31%	8.29%	1.99%	1.09%

从以上数据得出采用加权融合的策略进行融合效果更好，我们定义的融合公式见公式（5-1）所示：

$$fusionScore = \alpha \times porescore + \beta \times minuscore + (1 - \alpha - \beta) \times orientscore \quad (5-1)$$

式中 $porescore$ ——汗孔的匹配得分；

$minuscore$ ——细节点匹配得分；

$orientscore$ ——方向场的匹配得分。

5.3.2 同类算法对比

为了测试本文提出的两种算法的性能，我们分别与目前普遍应用于指纹识别的基于细节点的匹配算法，以及 A. K. Jain 提出的同样采用方向场，细节点，汗孔三种特征融合的高分辨率指纹匹配算法进行了对比实验。

数据库采用香港理工大学的 148 个手指的两个数据集 DBI 和 DBII，我们基于这个数据库做了如下实验：

第一组实验我们使用 DBI 数据集上的指纹数据进行了实验，匹配实验为使用 DBI 中的小图与小图进行两两匹配，测试残缺指纹间的匹配效果；

第二组实验采用数据集 DBII 中的大图进行了实验，匹配实验使用 DBII 中的大图两两匹配，测试完整指纹间的匹配效果；

第三组实验使用数据集 DBI 中的小图与数据集 DBII 中的大图进行匹配，测试残缺指纹与完整指纹间的匹配效果。

为了测试拒识率（FRR），我们选择 Genuine 集为每个指头的 10 张指纹图像两两互相匹配，这样就有 $148 \times 45 = 6660$ 次实验；为了测试误识率（FAR），我们选择

Impostor 集为每个指头在第二个时期采的第一张指纹图像与其他指头在第一时期采集的第一幅指纹图像，这样就有 $148 \times 147 = 21756$ 次实验。本文中最后得到的匹配分数全部归一化到 $(0, 1)$ 区间内。

基于以上的数据我们分别在三个数据集上做了如下的实验，仅用细节点特征的匹配实验，采用 A.K.Jain 提出的汗孔匹配加融合的实验，基于分块的汗孔匹配加融合的实验和基于脊线汗孔匹配加融合的实验。然后根据融合后的匹配结果从等误率和 ROC 曲线两个评价指标进行了对比。其中等误率定义为拒真率和认假率相等时的错误率，拒真率为把应该匹配的指纹认定为不匹配的概率，认假率为把不应该匹配的指纹认定为匹配的概率，衡量算法好坏的标准为等误率越低，整体性能越高。四种算法的等误率对比结果分别如表 5-4，表 5-5 和表 5-6 所示。四种算法的 ROC 曲线对比如图 5-3，图 5-4 和图 5-5 所示。通过 EER 的对比证明采用本文提出的两种匹配算法有效的提高了指纹匹配的准确率，相比同类算法准确率有所提高。

表 5-4 在数据集 DBI 上四种匹配方法的等误率对比

算法	正确识别个数	EER	ZeroFMR	ZeroFNMR
基于细节点的匹配	21411	24.65%	58.10%	63%
A.K.Jain 的匹配算法	21746	23.47%	53.83%	62%
基于分块的汗孔匹配后融合	22588	20.51%	31.50%	61%
基于脊线的汗孔匹配后融合	23074	18.80%	29.54%	61%

表 5-5 在数据集 DBII 上四种匹配方法的等误率对比

算法	正确识别个数	EER	ZeroFMR	ZeroFNMR
基于细节点的匹配	28025	1.376%	2.598%	100%
A.K.Jain 的匹配算法	28063	1.241%	2.447%	100%
基于分块的汗孔匹配后融合	28105	1.140%	1.757%	100%
基于脊线的汗孔匹配后融合	28125	1.089%	1.697%	100%

表 5-6 用数据集 DBI 上的小图匹配数据集 DBII 上的大图，四种算法等误率对比

算法	正确识别个数	EER	ZeroFMR	ZeroFNMR
基于细节点的匹配	22545	20.66%	74.5%	95%
A.K.Jain 的匹配算法	22784	19.82%	73.4%	94%
基于分块的汗孔匹配后融合	23053	18.87%	73.0%	93%
基于脊线的汗孔匹配后融合	23397	17.66%	71.2%	90%

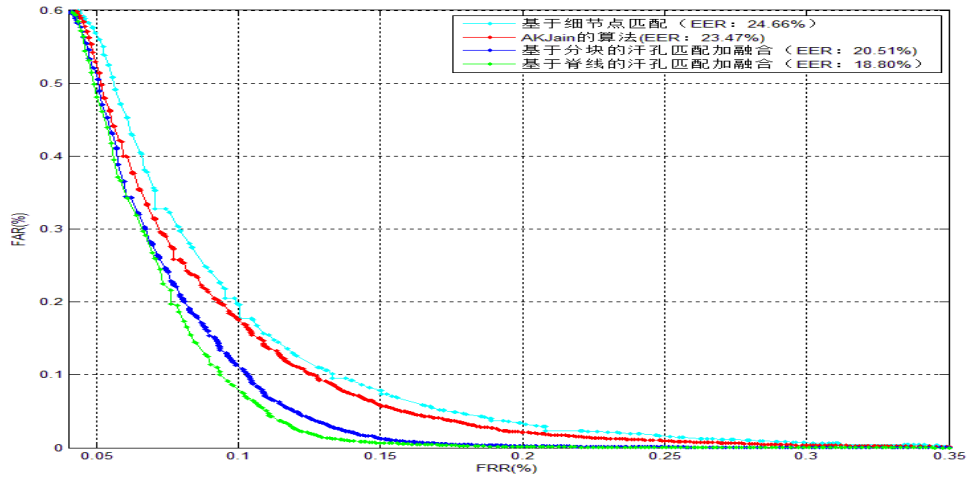


图 5-3 在 DBI 上四种算法的 ROC 曲线对比

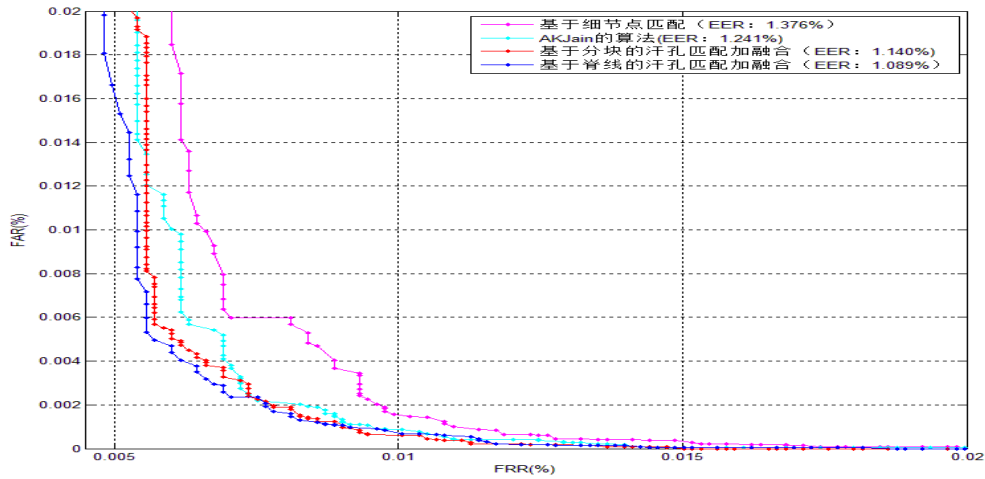


图 5-4 在 DBII 四种算法的 ROC 曲线对比

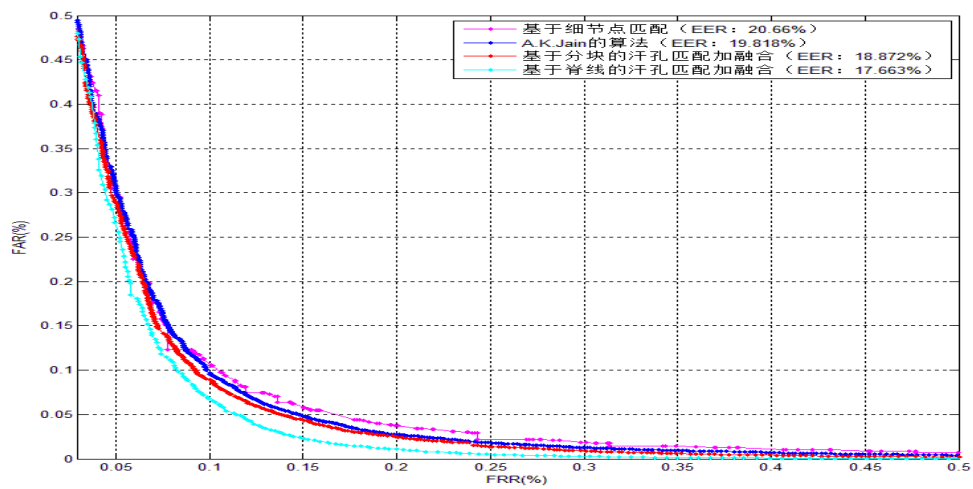


图 5-5 用 DBI 的小图匹配 DBII 上的大图的 ROC 曲线对比

通过在三个数据集上的 ROC 对比可见，基于脊线的汗孔匹配加融合的方法和基于分块的汗孔匹配加融合的算法都要比单独使用细节点匹配的效果好。相比 A.K.Jain 的方法本文提出的两种算法等误率（EER）更低。基于分块的汗孔匹配算法 EER 低于 A.K.Jain 的算法，与基于脊线的汗孔匹配算法相近，略高于基于脊线的汗孔匹配算法。

通过实验的结果表明，基于分块汗孔匹配后融合和基于脊线汗孔匹配后融合的算法能够有效提高指纹识别的性能，它们有如下优点：第一，使用多特征融合的方式，有效的提高了匹配的准确率和鲁棒性；第二，采用分块匹配的方式和按脊线匹配汗孔的方式，都是基于分而治之的思想，一定程度上解决了局部形变对匹配结果的影响；第三，相比 A.K.Jain 的算法，本文充分利用了大量的汗孔作为匹配依据，充分发挥了第三层指纹特征的优势，有利于高分辨率指纹匹配算法的发展。

5.3.3 效率对比

分析算法的复杂度，本文提出的两种算法相比同类算法都有一定的优势。为了验证本文提出两种算法复杂度低的优势，我们在同一台机器上做了匹配时间对比实验。机器的配置为 I3(2.93GHZ)双核处理器，内存为 4G，操作系统为 Windows 7。数据库同样采用香港理工大学高分辨率指纹库，数据库图像大小 640×480，规模 1480。选取 Genuine 集为每个指纹的第二个时期的每个指纹和第一个时期的每个指纹两两匹配共 $148 \times 25 = 3700$ 次匹配，Impostor 集为每个指纹第二时期采集的第一幅指纹图像和其他指纹图像第一时期采集的第一幅指纹做匹配，共 $148 \times 147 = 21756$ 次匹配。基于以上数据库，我们做了如下的实验，使用 Zhao 的匹配算法^[14]的匹配实验，基于细节点分块加融合的匹配实验，基于脊线的汗孔匹配加融合的匹配实验，对平均匹配时间做了对比，对比结果如表 5-5 所示。

表 5-7 平均匹配时间对比

算法	平均时间 (s)
直接汗孔匹配算法	19.53
基于脊线的汗孔匹配加融合	4.53
基于分块的汗孔匹配加融合	2.78

分析三种方法的平均匹配时间，我们可以看到，我们的平均匹配时间远低于直接汗孔匹配算法。原因在于我们提出的两种算法都采用分而治之的思想实现，

在配准和寻找匹配对的过程中，都采用分块处理的方式，所以时间复杂度大大降低。基于分块的汗孔匹配，根据已经匹配的细节点对图像进行分块，然后对每个小块内的汗孔进行匹配对比。相比对比整个图像的汗孔点，这种方法需要对比的汗孔数量规模大大下降，这样可以大大降低匹配的计算时间；另外在配准阶段，采用匹配的脊线上的点或者根据剖分后的三角形顶点进行配准参数的估计，相比搜索所有的汗孔点来估计配准参数，配准速度要快很多，而且更加准确。基于脊线的汗孔匹配算法，采用按照匹配脊线配准和匹配的方式，根据脊线的匹配结果匹配能够有效降低汗孔的匹配计算复杂度。首先脊线匹配直接对比的是一条条的脊线，按照每条脊线上的点进行对比。而 Zhao 的算法直接采用汗孔的匹配方法则计算复杂度较高，匹配汗孔分为两个步骤，第一步粗匹配要对比每一对汗孔之间的匹配得分，这意味要把两个指纹的所有汗孔都拿来计算一遍，相比本文中的两种算法计算规模要大很多，也容易出现误差。我们可以看到本文的两种算法平均匹配时间明显低于 Zhao 的方法，综合性能较高。

5.3.4 配准效果对比

基于高分辨率指纹图像的特点，被放大的局部形变对汗孔的误差影响变大，所以使用整体汗孔进行配准势必受到局部形变的影响，本文根据高分辨率指纹的这一特点，提出了基于细节点分块汗孔匹配和基于脊线的汗孔匹配两种方法。这两种算法在配准阶段，都采用局部配准的方式，能够很好的解决局部形变带来的误差，使最终配准的效果更好，从而提高匹配的准确率。本实验目的在于对比使用分块配准，基于脊线配准和整体逼近的方式配准的效果。实验数据为从 Genuine 集里随机挑选的 100 对指纹。对 100 对指纹，分别应用 ICP 算法，基于分块配准和基于脊线配准三种方式，对汗孔进行了配准。然后根据找到最终匹配的汗孔点，分别查看匹配汗孔点的配准效果，然后计算这些点在配准后的距离均方差。配准效果的对比如图 5-6 所示。

从图 5-6 可见，在一些局部地区，采用整体配准的方法并不理想，而采用本文中的两种局部配准方法，能够使配准结果更加准确。实验对比 100 对指纹配准后的平均配准距离均方差结果如表 5-8 所示。

表 5-8 对比采用整体 ICP 配准，按细节点分块配准和按脊线配准的效果

	基于 ICP 配准	基于分块配准	基于脊线配准
平均距离误差	46.9691	27.7423	27.0871

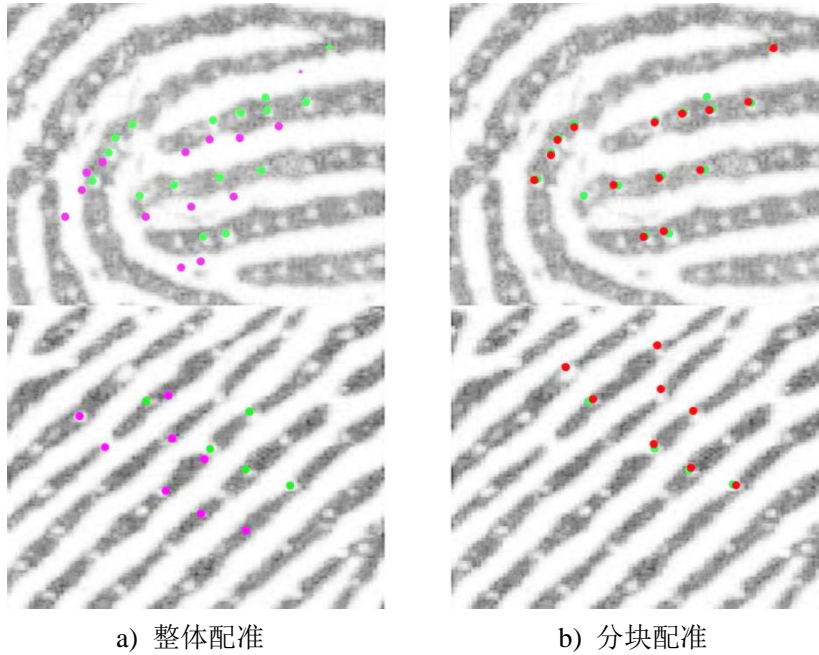


图 5-6 配准效果对比

从实验结果可以看出，采用分块配准的方式和采用基于脊线配准的方式比采用整体汗孔点逼近的配准方式效果更好。其中配准后相对应的汗孔的距离平方误差从 46.9691 下降到 27.7423 和 27.0871。说明采用本文提出的两种配准方式，使相应的汗孔点更接近于其要匹配的点。这样可以使最终的汗孔匹配更准确。

5.4 本章小结

本章对本文提出的基于分块汗孔匹配后融合和基于脊线汗孔匹配后融合的算法进行了匹配性能的测试实验。数据库使用香港理工大学的高分辨率指纹库。本文提出的基于细节点分块的汗孔匹配和基于脊线的汗孔匹配算法，都采用了“分而治之”的思想，对指纹进行区域配准，区域匹配，有效解决了局部形变对最终匹配结果的影响。本文提出的两种算法在实验中精度较高，相比同类算法 EER 较低，而且计算复杂度在同类算法中处于较低水平，说明本文提出的两种算法的综合性能较目前提出的高分辨率指纹识别匹配算法有所提高。

结 论

伴随着技术的进步，高分辨率指纹识别的研究也更加深入。汗孔等第三层指纹特征作为高分辨率指纹识别的主要应用，具有特征点多，防造假等优势，结合汗孔的指纹匹配能够有效提高指纹识别的精度，满足市场对指纹识别日益提高的要求。本文对高分辨率指纹识别的匹配做了深入研究，基于分治的思想提出了两种汗孔匹配算法，本文的主要内容有：

基于分治的思想，提出了一种基于分块的汗孔匹配算法，算法根据细节点对指纹进行分块，然后按照分块的结果，对每块内的汗孔分别进行匹配，最后与细节点和方向场等信息进行了融合。实验表明这种方法具有时间复杂度低，鲁棒性强，准确率高等优点，有效提高了高分辨率指纹匹配算法的性能。

提出了一种基于脊线的汗孔匹配算法，算法首先根据脊线进行匹配，再找到匹配的脊线后，再对脊线上相应的汗孔进行匹配，最后融合细节点和方向场等信息，得到了较高的准确率，并且由于最终得分包含了脊线，方向场，细节点，汗孔四种特征信息的匹配结果，鲁棒性较强。该算法的时间复杂度相对较低，综合性能有所提高。

本文提出的两种高分辨率指纹匹配算法性能较好，但是这两种算法仍有如下可改进之处：

基于细节点分块的汗孔匹配算法依赖于细节点的匹配效果，三角剖分是根据细节点的匹配点实现，如果细节点没有匹配，则对本算法的影响较大，造成的结果是拒真率较高；另外由于专注于理论的验证，所以算法的优化还有很大空间。

基于脊线的汗孔匹配依赖于脊线的匹配结果，关键在于找到准确的匹配脊线，由于脊线匹配的复杂性，算法中的脊线匹配算法部分尚有较大的改进空间。

参考文献

- [1] Galton F. Personal Identification and Description[J]. Nature, 1888, 38: 201-202.
- [2] Cummins H, Midlo M. Fingerprints, Palms and Soles: An Introduction to Dermatoglyphics[M]. Dover, 1961: 2-15.
- [3] Jain A K, Chen Y, Demirkus M. Pores and Ridges: High Resolution Fingerprint Matching Using Level 3 Features[J]. IEEE Trans, Pattern Analysis and Machine Intelligence, 2007, 29(1): 15-27.
- [4] Maltoni D, Maio D, Jain A K. A Handbook of Fingerprint Recognition[M]. Springer, New York, 2003: 1-56.
- [5] Ranade S, Rosenfeld A. Point Pattern Matching by Relaxation[J]. Pattern Recognition, 1980, 12(5): 269-275.
- [6] Ballard D H. Generalized Hough Transform to Detect Arbitrary Patterns[J]. IEEE Trans, Pattern Analysis and Machine Intelligence, 1981, 3(2): 111-122.
- [7] Stockman G, Kopstein S, Benett S. Matching Images to Models for Registration and Object Detection via Clustering[J]. IEEE Trans, Pattern Analysis and Machine Intelligence, 1982, 4(3): 229-241.
- [8] 漆远, 田捷, 邓翔. 基于遗传算法的指纹图匹配算法及应用[J]. 软件学报, 2000, 11(4): 488-493.
- [9] Vinod V V, Ghose S. Point Matching Using Asymmetric Neural Networks[J]. Pattern Recognition, 1993, 26(8): 1207-1214.
- [10] Sclaroff S, Pentland A P. Modal Matching for Correspondence and Recognition[J]. IEEE Trans, Pattern Analysis and Machine Intelligence, 1995, 17(6): 545-561.
- [11] Drygajlo K K, Morier P. Extraction of Level 2 and Level 3 Features for Fragmentary Fingerprints[C]. Proceedings of the Second Cost Action 275 Workshop, Vigo, Spain, 2004: 83-88.
- [12] Jain A K, Chen Y, Demirkus M. Pores and Ridges: Fingerprint Matching Using Level 3 Features[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(1): 15-26.
- [13] Ratha N K, Chen Y, Jain A K. Adaptive Flow Orientation Based Feature Extraction in Fingerprint Images[J]. Pattern Recognition, 1995, 28(10):

1657-1672.

- [14] Zhao Q, Zhang L, Zhang D, et al. Direct Pore Matching for Fingerprint Recognition[C], Proceedings of ICB'09, 2009: 97-606.
- [15] Liu F, Zhao Q, Zhang L, et al. Fingerprint Pore Matching Based on Sparse Representation[C]. Proceedings of the 20th International Conference on Pattern Recognition, 2010: 1630-1633.
- [16] Feng J J. Combining Minutiae Descriptors for Fingerprint Matching[J]. Pattern Recognition, 2008, 41(1): 342-352.
- [17] Hong L, Wan Y, Jain A K. Fingerprint Image Enhancement: Algorithm and Performance Evaluation[J]. IEEE Trans, Pattern Analysis and Machine Intelligence, 1998, 20(8): 777-789.
- [18] Maio D, Maltoni D. Direct Gray-scale Minutiae Detection in Fingerprints[J]. PAMI, 1997, 19(1): 27-40.
- [19] Chang J H, Fan K C. Fingerprint Ridge Allocation in Direct Gray Scale Domain[J]. Pattern Recognition, 2001, 34(10): 1907-1925.
- [20] Jiang X, Yau W Y, Ser W. Detecting the Fingerprint Minutiae by Adaptive Tracing the Gray-level Ridge[J]. Pattern Recognition, 2001, 34(5): 999-1013.
- [21] Almansa A, Lindeberg T. Neural Network Based Minutiae Filtering in Fingerprints[C]. ICPR98(SAP2), 1998: 1654-1658.
- [22] O’Gorman L, Nickerson J V. An Approach to Fingerprint Filter Design[J]. Pattern Recognition, 1989, 22(1): 29-38.
- [23] Sherlock B G, Monro M, Millard K. Fingerprint Enhancement by Directional Flourier Filtering[J]. IEEE Proceedings-Vision Image and Signal Processing, 1994, 141(2): 87-94.
- [24] Wu C Y, Zhou J, Bian Z Q, et al. Robust Crease Detection in Fingerprint Images[C]. Proceedings of CVPR, 2003: 505-510.
- [25] Huang D C. Enhancement and Feature Purification of Fingerprint Images[J]. Pattern Recognition, 1993, 26(11): 1661-1671.
- [26] Jain A K, Lin H, Bolle R. On-line Fingerprint Verification[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997, 19(4): 302-313.
- [27] Bhnu B, Tan X J. Learned Templates for Feature Extraction in Fingerprint Images Computer Vision and Pattern Recognition[C]. CVPR, Proceedings of the

- 2001 IEEE Computer Society Conference, 2001, 12: 591-596.
- [28] Coetzee L, Botha E C. Fingerprint Recognition in Low Quality Images[J], Pattern Recognition, 1993, 26(10): 1441-1460.
- [29] Liu J D, Huang Z, Chan K L. Minutiae Extraction from Gray-level Fingerprint Image by Relationship Examination[J]. IEEE, Image Processing, 2001, 2(9): 10-13.
- [30] Robby A R, Stosz J D. Fingerprint Features Statistical Analysis and System Performance Estimates[C]. Proceedings of IEEE, 1997, 85(9): 1390-1421.
- [31] Ray M, Meenen P, Adhami R. A Novel Approach to Fingerprint Pore Extraction[C]. Proceedings of Thirty-seventh Southeastern Symposium on System Theory, 2005: 282-286.
- [32] Zhao Q, Zhang L, Zhang D, et al. Adaptive Pore Model for Fingerprint Pore Extraction[C]. ICPR, 2008: 2833-2844.
- [33] 龚子桢, 花向红, 义崇政, 等. 基于 Delaunay 三角剖分的 ICP 算法研究与实现[J]. 测绘工程, 2010, 19(5): 29-31.
- [34] 陈兆林, 张书毕, 冯华俊. 基于 MATLAB 的二维坐标系变换[J/OL]. 中国科技论文在线, 2008, 1(6): 671-675, [2007-04-13]. <http://www.paper.edu.cn/index.php/default/releasepaper/content/200704-368>.
- [35] 邹乐强. 最小二乘法原理及其简单应用[J]. SCIENCE & TECHNOLOGY INFORMATION, 2010, 2: 282-283.
- [36] 李子奈, 叶阿忠. 高等计量经济学[M]. 清华大学出版社, 2000, 1: 27-29.
- [37] 黄铎, 陈兰平, 王凤. 数值分析[M]. 科学出版社, 2004, 3: 159-161.
- [38] Ross A, Jain A K. Information Fusion in Biometrics[J]. Pattern Recognition Letters, 2003, 4: 2115-2125.
- [39] Zhang D. Automated Biometrics: Technologies and Systems[M]. Norwell, MA: Kluwer, 2000: 23-84.
- [40] Jain A K, Flynn P, Ross A. Handbook of Biometrics[M]. Springer, New York, 2008: 271-291.
- [41] Tax D M, Breukelen M, Duin P W, et al. Combining Multiple Classifiers by Averaging or by Multiplying[J]. Pattern Recognition, 2000, 9: 1475-1485.
- [42] Xu Y, Zhu Q, Zhang D, et al. Combine Crossing Matching Scores with Conventional Matching Scores for Bimodal Biometrics and Face and Palmprint

- Recognition Experiments[J]. Neurocomputing, 2011, 74: 3946-3952.
- [43] Feng J J, Ouyang Z Y, Cai A. Fingerprint Matching Using Ridges[J]. Pattern Recognition, 2006, 39: 2131-2140.
- [44] Hrechak A K, McHugh J A. Automatic Fingerprint Recognition Using Structural Matching[J]. Pattern Recognition, 1990, 23(8): 893-904.
- [45] Wahab A, Chin S H, Tan E C. Novel Approach to Automated Fingerprint Recognition[C]. Proceedings of IEE Visual Image Signal Processing, 1998: 160-166.
- [46] Pdanti S, Prabhakar S, Jain A K. On the Individuality of Fingerprints[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 8(24): 1010-1025.
- [47] Senior A. A Hidden Markov Model Fingerprint Classifier[C]. Proceedings of the Thirty-First Asilomar Conference on Signals, Systems & Computers, 1997: 306-310.

哈尔滨工业大学学位论文原创性声明及使用授权说明

学位论文原创性声明

本人郑重声明：此处所提交的学位论文《基于汗孔的高分辨率指纹匹配算法》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果。据本人所知，论文中除已注明部分外不包含他人已发表或撰写过的研究成果。对本文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。本声明的法律结果将完全由本人承担。

作者签名： 方博

日期： 2012 年 1 月 7 日

学位论文使用授权说明

本人完全了解哈尔滨工业大学关于保存、使用学位论文的规定，即：

(1) 已获学位的研究生必须按学校规定提交学位论文；(2) 学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文；(3) 为教学和科研目的，学校可以将学位论文作为资料在图书馆及校园网上提供目录检索与阅览服务；(4) 根据相关要求，向国家图书馆报送学位论文。

保密论文在解密后遵守此规定。

本人保证遵守上述规定。

作者签名： 方博

日期： 2012 年 1 月 7 日

导师签名： 徐勇

日期： 2012 年 1 月 7 日

致 谢

在此我要向在我攻读硕士研究生阶段，给予我指导，关心，帮助的所有老师，亲人，同学和朋友表示我由衷的感谢。

本论文的顺利完成，离不开我的导师徐勇教授的关心和帮助。从我的课题选择，到论文的顺利完成，徐老师孜孜不倦的指导与教诲，提供了莫大的帮助。徐勇老师多次询问我的研究进程，并帮助我开拓思路，精心点拨，热忱鼓励。徐勇老师一丝不苟的作风，严谨求实的态度，踏踏实实的精神，渊博的学识，对专业孜孜以求，精益求精的热忱，无时无刻不在鼓励前进。如果说我从徐老师那里学会了怎样做好学问，那么首先应该说我从徐勇老师那里领略了真正的学术精神，徐勇老师严谨的治学态度和坚韧的探索精神将使我终生受益。

衷心地感谢卢光明副教授，在我的课题研究过程中，他一直给予我热心的帮助，为我的研究工作提出了很多行之有效的改进意见。

衷心地感谢我的师姐刘凤的鼎力支持，她为这篇论文的完成提供了巨大的帮助，她认真的工作态度和耐心的指导，深深的感染我向她学习。在此向她表示最真诚的谢意。

衷心地感谢哈工大深圳研究生院的全体老师。感谢他们为我提供了一个优秀的学习生活环境，是他们慷慨地为我授业解惑，把我领进了一个深奥而又迷人的殿堂。感谢亲爱的师兄弟们，感谢他们在科研工作中给予的大力帮助！

衷心地感谢我的家人多年来对我的无私支持，在我遇到困难时，为我分忧解难，在我迷惑时，为我指引方向，使我在通往知识的大路上没有顾虑奋勇向前，焉得原草，言树之背，养育之恩，无以言报，父母永远健康快乐是我最大的心愿。

感谢百忙之中评阅论文和参与答辩的各位老师！