

基于奇异点邻近结构的快速指纹识别*

时鹏^{1,2}, 田捷^{1,2+}, 苏琪^{1,2}, 杨鑫^{1,2}

¹(中国科学院 自动化研究所 复杂系统与智能科学重点实验室 生物识别与安全技术研究中心,北京 100190)

²(中国科学院 研究生院,北京 100049)

Fast Fingerprint Identification Based on Neighborhood Structure Around Singular Point

SHI Peng^{1,2}, TIAN Jie^{1,2+}, SU Qi^{1,2}, YANG Xin^{1,2}

¹(Center for Biometrics and Security Research, Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, The Chinese Academy of Sciences, Beijing 100190, China)

²(Graduate University, The Chinese Academy of Sciences, Beijing 100049, China)

+ Corresponding author: E-mail: tian@doctor.com

Shi P, Tian J, Su Q, Yang X. Fast fingerprint identification based on neighborhood structure around singular point. *Journal of Software*, 2008,19(12):3134-3146. <http://www.jos.org.cn/1000-9825/19/3134.htm>

Abstract: Combining the classification and matching of fingerprints together, a neighborhood structure is proposed in this paper, which includes the orientation field and minutia around the reference singular point. This structure has the advantage that the identification information is centralized around the singular point, and can dramatically decrease the calculation of matching. It can also be directly used as pattern in both the continuous classification and the fast matching of fingerprints, and carry out the fast identification of the large scale database. Experimental results on NIST and FVC2004 databases show that this algorithm can highly speed up the matching of large scale fingerprint database with a preferable performance, and it can be used in one-to-many matching of on-line fingerprint identification system.

Key words: fingerprint identification; classification; matching; singular point; minutia pair

摘要: 将指纹识别中分类和匹配过程相结合,提出了一种包含奇异点周边的方向场和细节点等特征的奇异点邻近结构.该结构利用奇异点周边识别信息集中的特点,大大减少了匹配的计算量,并能够同时作为指纹分类和比对的特征,直接应用于指纹的连续分类和快速匹配过程,实现对大容量指纹数据库的快速识别.在 NIST 和 FVC2004 数据库上的测试结果显示,该算法在保证自动指纹识别系统(automatic fingerprint identification system,简称 AFIS)的识别准确性的同时,还使得指纹在线识别系统的 1:N 辨识速度有显著的提高.

关键词: 指纹识别;分类;匹配;奇异点;细节点对

中图法分类号: TP301 文献标识码: A

指纹是目前在生物特征识别领域应用最为广泛的生物特征之一.由于指纹具有唯一性和不变性等优良特

* Supported by the National Natural Science Foundation of China under Grant Nos.60875018, 60575007 (国家自然科学基金); the Chinese Academy of Sciences Hundred Talents Program (中国科学院百人计划)

Received 2007-02-06; Accepted 2007-08-24

性,并且便于采集、抗干扰性强,使得自动指纹识别系统在身份鉴别领域得到了广泛的应用.指纹识别算法从运行方式而言,可以分为两类,即 1:1 验证和 1:N 辨识.1:1 验证是把一个现场输入的指纹和一个已经注册的指纹进行一对一的比对,以确认指纹持有者的身份;而 1:N 辨识是把现场采集到的指纹与数据库中的指纹相比对,从中找到与现场输入的指纹相匹配的指纹.对于大容量的指纹数据库,传统的 1:1 验证算法在执行效率方面不能满足现场比对的需要,因此必须采用兼顾识别率和识别速度的 1:N 快速指纹识别算法,以符合自动指纹识别系统实用化的需求.

针对大容量指纹数据库的应用所带来的快速指纹识别问题,通常的解决方式是保持指纹匹配算法不变,而通过分类算法把数据库中的所有指纹分成固定的类别,以减少识别过程中检索指纹的数量,从而提高系统运行的速度.根据分类体系的不同,指纹分类可以分为固定类别分类方法和连续分类方法.常见的固定类别分类方法是根据指纹的自然形状将指纹分为 5 类,或扩展为 6~8 类,这种分类体系称为 Henry 分类体系^[1].Henry 分类体系中重要的分类特征有指纹的奇异点,包括 Core 点(中心点)和 Delta 点(三角点),以及指纹方向场和脊线流等.Chang 等人^[2]提出了一种脊线流分布模型用于指纹分类,定义了 10 种指纹脊线常见的形状模型,通过分析脊线的形状和分布来确定指纹所在的类别.这种方法的缺点是脊线模型基于先验知识定义,抗形变能力不强.Tan 等人^[3]提出了基于特征学习的指纹分类算法,利用遗传算法计算指纹的组合分类特征.这类基于学习的分类算法仍然存在着计算复杂度高、训练时间长等不足.另一方面,基于 Henry 体系的分类算法存在一个共同的问题,即只能将指纹数据库分成 5 类左右,并且由于这种指纹分类方法遵循自然分布的规律,而左旋、右旋和漩涡类指纹分别占到了指纹自然分布的 33.8%,31.7%和 27.9%,其他两类包括拱形和尖拱形,仅占到总数的 3.7%和 2.9%^[1],分布极不均衡,因此不能有效地减少系统运行时的搜索量,加速效果并不明显.

连续指纹分类方法是通过一种相似度变换生成概括了指纹主要特征的分类特征向量.检索时,将输入的指纹分类特征在多维空间中与指纹模板中的特征向量进行计算,找到与其对应特征向量相近的一些指纹作进一步的匹配.已有的连续指纹分类方法大部分是采用指纹方向场作为分类特征.Lumini 等人^[4]最早提出连续分类概念,使用从已配准的指纹方向场数据中提取出来的分类特征向量,利用最近邻方法等相似性度量方法搜索最相似的备选指纹模板并作进一步的匹配.Cappelli 等人^[5]提出了基于预配准的分块指纹方向场信息的方法.该方法融合基于 Dynamic Mask(MASK)和 Multi-Space Karhunen-Loeve Transform(MKL)的分类方法进行连续指纹分类,通过在 NIST DB14 数据库上测试,在检索范围为 10%的情况下,检索准确率为 86.8%.Boer 等人^[6]提出了融合 3 种不同指纹特征进行连续指纹分类的方法.该方法融合了配准的指纹方向场信息、FingerCode 特征和细节点三角结构特征,最终采用多分类器融合的方法进行连续指纹分类.在 FVC2000 DB2 数据库上的实验表明,多特征融合比单一特征的指纹分类效果要好,在检索范围为 10%的情况下,检索准确率约为 98%.Bhanu 在文献[7]中提出了一种基于细节点三角结构的连续指纹分类方法.该方法利用细节点三角结构提取边夹角、最大边长、三角形结构类型等具有旋转、平移不变性的特征作为分类特征向量.利用指纹匹配的方法和固定阈值技术在指纹数据库中搜索相似的指纹模板.Li 等人^[8]提出了使用复数滤波器对已配准的指纹图像进行滤波,提取奇异点的相对位置信息,并使用欧氏距离进行指纹间相似度的计算.使用 NIST DB4 数据库测试,在检索范围为 12.5%的情况下,检索准确率为 90%.然而,上述方法也存在一些共同的问题,大部分指纹分类算法都使用经过旋转、平移配准的指纹方向场信息作为分类特征,而对于现场采集的指纹,提取指纹主方向进行旋转配准工作存在一定的难度,很容易出现配准错误,进而影响指纹检索的准确率.

综合以上研究,连续分类方法能够避免指纹类别难以确定、子类中的指纹数量过大等缺点,并且能够通过调整分类邻域的范围来平衡系统的精度与速度.然而,通常的连续分类方法往往只侧重于数据库分类方面的研究,而对于决定系统运行时间的另外一个因素,即指纹匹配速度的提高缺乏关注,没有将二者结合起来考虑.这样,指纹辨识中分类和匹配之间彼此孤立,不能充分发挥 1:N 算法在速度上的优势.因此,许多在小数据库上有良好表现的识别算法,并不能直接应用在大容量数据库的指纹识别上.

针对这些情况,我们提出了一种基于奇异点邻近结构的快速指纹识别算法.该算法通过构建围绕奇异点的邻近结构,把奇异点周边的方向场和细节点等特征统一考虑.该结构既可以用于指纹的连续分类,又可以用于模

板的快速匹配,能够把指纹的分类和匹配过程有效地结合起来.如图 1 所示,该算法结合了连续分类和快速匹配算法的优点,由于连续分类能够有效地减少指纹在数据库中的检索范围,因此 1:N 辨识时需要检索的指纹数量大为减少.与此同时,由于快速指纹匹配所需要的特征已在构建邻近结构的过程中计算出来,指纹匹配的时间也大大加快了,并且匹配的正确率仅有极小的降低.通过构建奇异点邻近结构,并通过分类和匹配等步骤,完成了整个 1:N 快速识别过程.

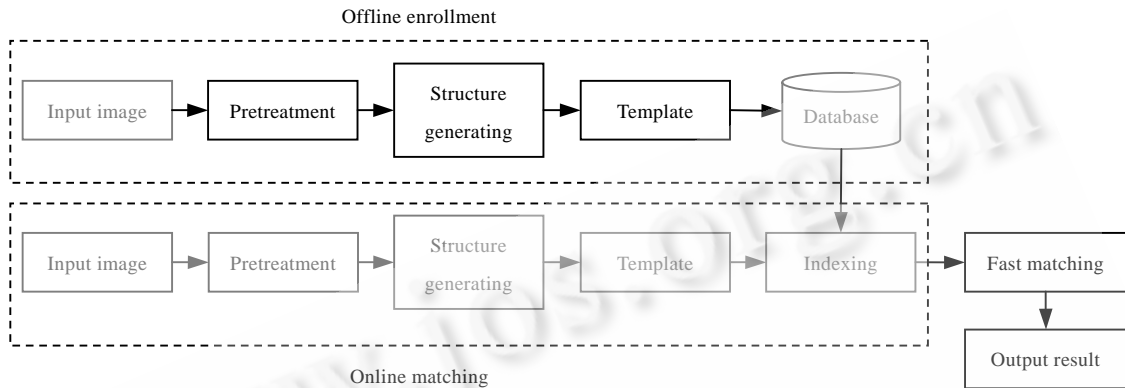


Fig.1 Process of 1:N fast fingerprint identification

图 1 1:N 快速指纹识别流程

本文第 1 节首先给出奇异点邻近结构的构造方法,并介绍利用该邻近结构进行用于连续分类的方向场特征提取算法,以及结构中用于快速匹配的细节点集提取算法.第 2 节给出在线进行 1:N 辨识的算法流程,包含基于奇异点邻近结构的连续分类和快速匹配算法.第 3 节给出在 NIST DB4^[9]和 FVC2004^[10] DB1 数据库上的实验结果.第 4 节对算法进行总结.

1 奇异点邻近结构

为了进行基于方向场信息的连续分类,并获取最有效的细节点特征,加速匹配过程,本文提出了一种新的奇异点邻近结构,并给出在邻近结构中进行指纹方向场特征提取和细节点集构造的算法.构建该结构的流程如图 2 所示.该算法首先对指纹图像进行预处理,并提取指纹中的奇异点,选取 Core 点作为参考点,在参考点周围构建以同心圆和网格组成的邻近结构.这种新的结构利用 Core 点周围识别信息集中的特点,将方向场信息和细节点信息统一在同一个结构中进行计算,并应用在指纹分类和匹配过程中.与传统的基于 Core 点的结构相比,具有以下几项优点:

首先,奇异点的相关信息应用在匹配过程中,将奇异点特征和细节点特征相结合,可以使算法同时具有基于全局特征和基于局部特征匹配算法的优点,提高算法的鲁棒性.

其次,如图 2 所示,在构建结构的过程中,需要同时提取方向场特征和细节点特征,其中方向场特征用于连续分类,而细节点特征用于快速匹配,这样使得分类和匹配不再是彼此独立的过程,而是有机地结合起来.

最后,由于指纹在 Core 点周围的细节点最为集中,在选取了基于奇异点的结构之后,区域内包含的识别信息可以有效地进行指纹匹配,而无须用到指纹中所有的细节点.一方面可以缩小模板的大小,使数据库能够存储更多的指纹模板;另一方面可以显著降低匹配运行的时间,提高系统运行的速度.

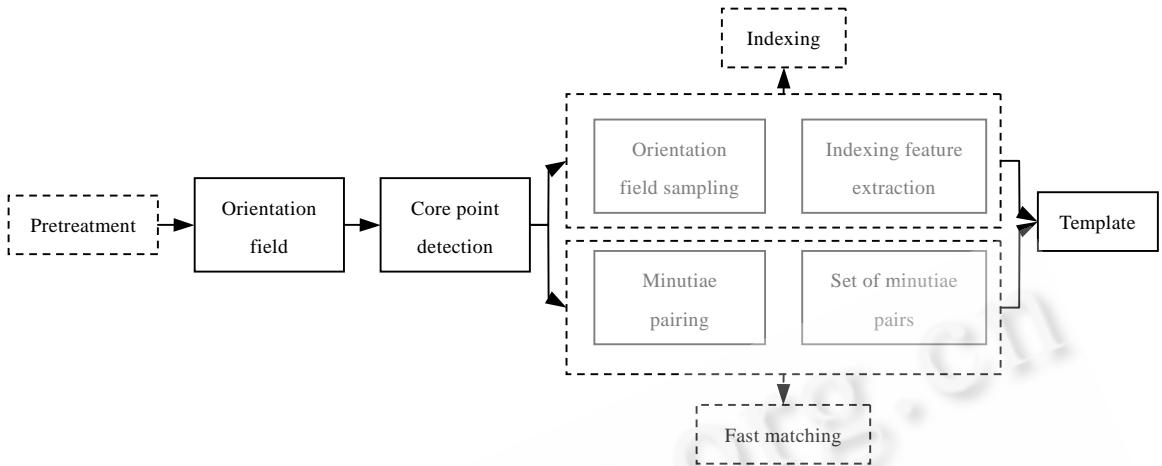


Fig.2 Process of construction of neighborhood structure around singular point and fingerprint template generation
图 2 构建奇异点邻近结构和指纹模板生成流程

1.1 指纹图像预处理

在构建奇异点邻近结构之前,首先需要对指纹图像进行预处理.指纹图像的预处理过程包括方向场计算、图像分割、二值化、细化以及细节点提取等步骤,我们采用Hong等人^[11]提出的最小平均方差方法计算出指纹的方向场信息,该方法在计算复杂度和精度上有较好的平衡,估计的方向场图像比较平滑.然后,采用罗希平等^[12]提出的方法对指纹图像进行二值化处理并提取细节点.根据方向场对指纹图像进行二值化处理,图像背景和谷线的像素灰度值为 255,脊线像素的灰度值为 0.最后,应用Kawagoe等人^[13]提出的Poincare Index方法进行奇异点检测.Poincare Index是最常用的奇异点检测算法,是将指纹图像分为固定的块,基于每个像素点进行局部纹线方向的计算,然后计算每块的块方向,最后以块方向为计算单位计算每一个像素的Poincare Index值,能够有效地检测得到指纹奇异点的位置和类型,包括Core点和Delta点.在计算过程中,采用多分辨率处理方法,分别使用 3×3 和 5×5 模板提取Core点,以提高参考点提取的准确度.当指纹图像中存在一个以上的Core点时,选择位于指纹图像上方的Core点作为参考点.当指纹类型为拱形等不存在Core点的类型时,特征提取算法使用Liu等人^[14]提出的参考点选取方法,提取指纹图像中曲率变化最大的位置作为参考点.

指纹方向场信息真实地反映了指纹图像的纹理特征,是对指纹纹线形状的一种全局的、抽象的描述,因此,大多数指纹分类算法都采用指纹方向场信息作为分类特征之一.指纹方向场特征提取算法的主要步骤是:

首先,使用基于梯度的方法求出基本指纹方向场数据.为了降低指纹方向场数据受图像质量的影响,并提高计算的速度,在实际运算过程中采用了滑动窗口技术.

$$\theta(i, j) = \frac{1}{2} \tan^{-1} \left(\frac{\theta_y(i, j)}{\theta_x(i, j)} \right), \begin{cases} \theta_x(i, j) = \sum_{u,v \in B_1} (2\partial_x(u, v)\partial_y(u, v)) \\ \theta_y(i, j) = \sum_{u,v \in B_1} (2\partial_x(u, v)\partial_x(u, v)) \end{cases} \quad (1)$$

其中, $\theta(i, j)$ 表示输入指纹图像 I_m 中的像素点 (i, j) 的方向. B_1 表示指纹图像块窗口, 实验中采用 15×15 像素的分块窗口. $\partial_x(u, v)$ $\partial_y(u, v)$ 分别表示像素 (u, v) 处的 x, y 方向导数.

由于指纹图像中存在噪声、奇异点等低质量的区域,对求出的指纹方向场数据需要进行低通滤波平滑.

$$o(i, j) = \frac{1}{2} \tan^{-1} \left(\frac{\theta'_y(i, j)}{\theta'_x(i, j)} \right) \quad (2)$$

$$\begin{cases} \theta'_x(x, y) = \sum_{u,v \in B_2} (h(u, v)\phi_x(i - uW, j - vW)) \\ \theta'_y(x, y) = \sum_{u,v \in B_2} (h(u, v)\phi_y(i - uW, j - vW)) \end{cases} \begin{cases} \phi_x(i, j) = \cos(2\theta(i, j)) \\ \phi_y(i, j) = \sin(2\theta(i, j)) \end{cases} \quad (3)$$

其中, $h(u, v)$ 为低通滤波函数, 实验中滤波窗口 B_2 为 20×20 . $o(i, j)$ 表示像素点 (i, j) 经过平滑后的方向数据, 其变化范围为 $0 \leq (i, j) < \pi$. 图 3 为一幅分割后的指纹图像及其方向场图像示例.

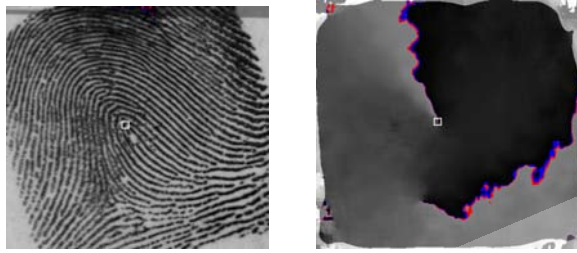


Fig.3 Fingerprint orientation field image (after segmentation) samples

图 3 指纹及其方向场图像(分割后)示例

1.2 用于连续分类的结构

在建立的奇异点邻近结构中包含两部分, 一部分用来提取连续分类特征, 另一部分则用于快速匹配特征提取. 首先, 给出整体奇异点邻近结构的构造方法, 并介绍分类特征提取的步骤. 如图 4 所示, 奇异点邻近结构以参考点为圆心, 以 r_k 为半径作 n 个同心圆, 其中 r_k 是预定义的半径值, 单位为像素的个数. 通过 n 个同心圆环, 可以将奇异点的邻近区域分为 n 个子区域, 第 k 个圆环称为第 k 档. 其中, 实线网格部分是用于提取分类特征的部分, 而所有的同心圆环, 包含虚线圆环, 是用于构造对比特征的部分. 实线和虚线同心圆的数量根据不同指纹数据库中图像的性质, 通过实验找到最优值. 根据上述规则, 就构成了用于指纹连续分类和快速比对的奇异点邻近结构.



Fig.4 Neighborhood structure around singular point

图 4 奇异点邻近结构

指纹分类特征中方向场采样点的分布方式如图 4 所示. 以指纹参考点为中心, 在指纹方向场图像中使用以 r_k 为半径, 以 $\Delta\theta$ 为角度间隔的极坐标表示方法构成采样点网格结构, r_k 的间隔为 m 个像素.

指纹方向场采样点构成采样点向量 $\psi = \{\psi_k | \psi_k = o(r_i, \theta_i), k=1, 2, \dots, N, \theta_i \in [0, 2\pi]\}$. 由于 NIST DB4 和 FVC2004DB1 数据库中的指纹图像面积不同, 我们将指纹奇异点邻近结构最大半径 r_n 分别设置为 200 和 150. 经过实验, 当 $m=20$, $\Delta\theta=\pi/8$ 时, 可以在保证分类算法性能的前提下, 程序运行速度最快. 此时, NIST DB4 和 FVC2004 DB1 中指纹图像提取的方向场采样点向量的维数分别为 168 和 117.

1.3 用于快速比对的结构

上述的奇异点邻近结构将参考点周边区域用网格进行方向场采样, 获得指纹的分类特征. 与此同时, 在结构中, 参考点周边的细节点也被该结构划分为不同的区域之中. 利用指纹局部结构的稳定性, 对结构中的细节点按照一定规则进行配对, 从而生成指纹模板, 用于快速匹配过程中.

如图 5 所示, 细节点在图中以小圆点表示, 结构中的细节点分布在不同的同心圆环区域内. 在进行方向场采样时, 邻近结构被同心圆和从参考点引出的射线划分为许多网格. 而在生成快速比对的细节点对时, 细节点被 k

个同心圆环分为不同的档次 ℓ , 这样奇异点邻近区域中的细节点就分布在 k 个分档之中, 作为构成指纹模板的主要特征. 为了获得更多的细节点信息, 需要在方向场采样所使用的邻近结构的基础上, 再向外扩展一些同心圆, 如图 4 中指纹图像上的虚线圆环部分所示, 这样可以使尽可能多的细节点包含在结构中, 对于保证算法的匹配正确率起着重要的作用.

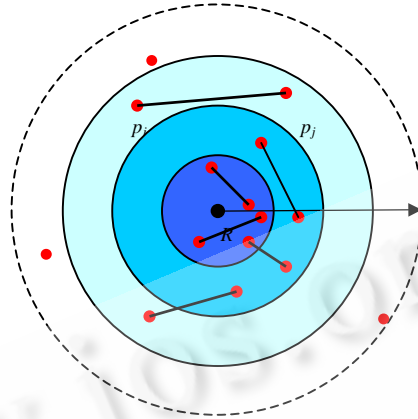


Fig.5 Selection of minutia pairs in the neighborhood structure around singular point

图 5 奇异点邻近结构中细节点对的选取

图 5 中, R 为参考点, p_i 和 p_j 分别为两个细节点, 连线表示这两个细节点可以构成一个细节点对. 在该结构中, 细节点矢量的 \bar{p}_i 定义为

$$\bar{p}_i = [(x, y), \theta, \ell, ZOD] \tag{4}$$

$$\ell = \frac{|p_i - R|}{r} \tag{5}$$

其中, (x, y) 是细节点 p_i 在全局中的坐标, θ 是 p_i 的方向, ZOD 是该细节点的方向场信息, ℓ 是细节点 p_i 所在的分档号, 对于第 k 档的细节点, 其分档号 $\ell = k$, 对于没有检测到奇异点的指纹, 默认 ℓ 值为 0. 对邻近结构中细节点分档的目的是为了选取最优的细节点对, 在保证匹配精度的同时, 减少模板的容量, 加快匹配的速度.

2 指纹在线识别

指纹在线识别的过程是在数据库里存储的指纹模板中, 找到与输入指纹相同的指纹模板. 由于在离线过程中已将数据库中的指纹进行了分类, 因此, 在线比对的第 1 步就是要找到与输入指纹相同和相邻近类别, 然后对类内的模板进行一一比对, 从而找到真匹配的指纹模板, 完成身份认证过程. 在本算法中, 为了进一步缩小查询范围, 我们采用了基于奇异点邻近结构内的方向场特征的连续分类. 在完成对输入指纹的连续分类之后, 对同类和相邻两类的指纹模板逐一进行 1:1 的基于点集的快速匹配, 完成整个 1:N 指纹辨识过程.

2.1 分类特征计算

对于在奇异点邻近结构中提取到的方向场特征, 采用指数方程对采样得到的方向场数据进行变换, 使得指纹方向场信息与指纹旋转角度相独立. 然后, 使用基于相关距离的相似度计算方法计算两枚指纹的分类特征间的距离, 对指纹进行分类. 由于指纹分类特征中采用的基于奇异点邻近结构的指纹方向场信息无须进行预配准, 该相似度计算方法极大地降低了指纹分类特征提取所需的时间, 同时减少了因指纹配准引入的分类错误.

由于指纹图像在采集的过程中都会出现不同程度的旋转, 使得像素点 (i, j) 处的真实指纹方向场信息 $\phi_{i,j}$ 都增加了一个手指旋转角度的信息 φ , 即 $\phi_{i,j} = \phi_{i,j} + \varphi$. 使用混合的指纹方向场信息 $O_{i,j}$ 对于指纹的分类和匹配操作都增加了很多不必要的麻烦. 现有的指纹分类方法都需要将混合的指纹方向场特征进行预配准 (包括旋转、平移), 对分类特征的配准效果直接影响分类的可靠性. 本文使用复数空间映射, 对方向场采样向量进行处理, 提取分类特征向量.

由于方向场采样点的取值范围是 $0 \leq \psi_k < \pi$, 角度变换不具有连续性, 在提取分类特征之前, 先将其扩展到 $0 \leq 2\psi_k < 2\pi$. 然后在复数空间进行特征映射, 得到分类特征向量.

$$V = \{v_k \mid v_k = e^{i2\psi_k} = \cos(2\psi_k) + i \cdot \sin(2\psi_k), k = 1, 2, \dots, N\} \quad (6)$$

本文为了减少指纹图像配准过程对特征分类的计算带来的误差, 使用基于特征向量间相关距离的相似度计算方法. 分类特征距离的计算公式如下:

$$d(v^q, v^t) = \frac{\frac{1}{N} \sum_{k=1}^N (v_k^q - \bar{V}^q)(v_k^t - \bar{V}^t)}{\sigma^q \cdot \sigma^t} \quad (7)$$

其中, v^q, v^t 表示 N 维输入特征向量和模板特征向量, \bar{V} 和 σ 分别表示相应特征向量的均值和方差. \bar{V} 和 σ 的计算公式分别为

$$\bar{V} = \frac{1}{N} \sum_{k=1}^N v_k \quad (8)$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{k=1}^N (v_k - \bar{V})^2} \quad (9)$$

由于在复数空间中可以将分类特征向量中附加的手指旋转角度信息 ϕ 与真实指纹方向场信息 $\phi_{i,j}$ 分离, 即 $e^{i\psi_k} = e^{i\phi_k} \cdot e^{i\phi}$. 分类特征向量可以表示为

$$\begin{aligned} V &= U \cdot e^{i2\phi}, \\ U &= \{u_k \mid u_k = e^{i2\phi_k}, k = 1, 2, \dots, N\} \end{aligned} \quad (10)$$

将分离出手指旋转角度信息的分类特征向量 V 带入公式(7)中, 可得经过旋转和平移变换后的分类特征距离计算公式:

$$\begin{aligned} d(V^q, V^t) &= \frac{\frac{1}{N} \sum_{k=1}^N (v_k^q - \bar{V}^t)(v_k^t - \bar{V}^t)}{\sigma_v^q \cdot \sigma_v^t} \\ &= \frac{\frac{1}{N} \sum_{k=1}^N (u_k^q - \bar{U}^t) \cdot e^{i2\phi} \cdot (u_k^t - \bar{U}^t) \cdot e^{i2\phi}}{\sigma_u^q \cdot e^{i2\phi} \cdot \sigma_u^t \cdot e^{i2\phi}} \\ &= d(U^q, U^t) \end{aligned} \quad (11)$$

上式说明, 虽然在分类特征距离计算公式中使用了混合指纹方向场信息 $O_{i,j}$, 但是最终在指纹相似度计算过程中起作用的仍然是真实指纹方向场信息 $\phi_{i,j}$.

2.2 细节点对的选取

传统的点对点的匹配算法存在着抗干扰性不强的问题, 很容易受到指纹图像的质量和形变等因素的影响, 因此, 需要将细节点按照一定的规则进行组合, 形成一定的集合进行匹配, 以增强匹配算法的鲁棒性. 这样可以减少噪声和形变对匹配算法性能的影响, 同时也可以提高匹配的准确率. Lee 等人^[15]以细节点对的集合作为主要特征, 提出了一种鲁棒的基于细节点对的指纹配准算法, 配准后的点对集可以直接用于比对. 然而, 许多细节点配对算法都遇到了一个共同的问题, 由于细节点数目较大, 而细节点对的数量又与细节点数的平方成正比, 数量繁多的细节点对使得模板变得很大, 比对速度也随之下降. 本文中, 我们利用奇异点邻近结构对细节点进行分档归类, 再根据一定的匹配规则生成细节点对集. 该算法一方面可以极大地减少模板中细节点对集的数量, 同时减少模板所占用的存储空间, 另一方面能够提取到最有效的匹配信息, 减少了匹配算法在点对匹配时的计算量, 使匹配速度得到了显著的提高.

在以点对为主要比对特征的算法中, 点对的数量对于算法性能有着重要的影响. 理想情况下, 点对匹配数量 Num 的计算公式为

$$Num = M(M-1) \times N(N-1) / 4 \quad (12)$$

其中, M 和 N 分别为输入和匹配对象指纹的细节点数. 从公式(12)可知, 在这种情况下, 会产生许多无用的或错误

的点对,不仅占用了存储空间,而且造成点对集匹配时查询时间的浪费.在本文的快速匹配算法中,利用细节点在邻近结构中的分布作为细节点对的选取规则.假设 P_i 和 P_j 是输入指纹的细节点集, Q_i 和 Q_j 是匹配对象指纹的细节点集.细节点对 $\overline{p_i p_j}$ 和 $\overline{q_i q_j}$ 按照如下规则选取:

$$\begin{cases} \overline{p_i p_j} | \ell_p^i - \ell_p^j \leq 1 \\ \overline{q_i q_j} | \ell_q^i - \ell_q^j \leq 1 \end{cases} \quad (13)$$

其中, $\ell_p^i, \ell_p^j, \ell_q^i$ 和 ℓ_q^j 分别为细节点 p_i, p_j, q_i 和 q_j 在邻近结构中所在的分档号.由于细节点已经通过分档号归为一定的类别,因此根据上述规则,可能构造的细节点对的数量就会大为减少,在匹配过程中对细节点对的检索过程也随之得到显著的加速.

2.3 基于点集的快速匹配

在选定了待匹配点集之后,可以顺利地进行配准和匹配过程.我们采用了陈宏等人^[16]提出的基于检验配准模式的配准方法,并对Jiang等人^[17]提出的基于队列的匹配算法进行了改进,应用在本文的快速匹配之中.算法步骤如下:

1.定义每个细节点矢量为 $\overline{p_i}$,每个指纹的特征中包含了所有的细节点矢量.由于细节点矢量结构属于指纹局部特征,因此细节点矢量集在整体上表现为较好的可区分性.

2.在预比对中,应用细节点矢量获得待比对的细节点对,如图6所示,其结构如公式(14)所示:

$$Fl_k = (d_{k_i}, d_{k_j}, \theta_{k_i}, \theta_{k_j}, Z_{k_i}, Z_{k_j}, n_{k_i}, n_{k_j}, t_{k_i}, t_{k_j})^T \quad (14)$$

该结构表示为多维矢量的形式,其 d_{k_i}, d_{k_j} 分别是细节点 k 和细节点 i, j 之间的欧氏距离; z_{k_i}, z_{k_j} 分别是 k 和 i, j 之间的方向差; $\theta_{k_i}, \theta_{k_j}$ 是 $\overline{k_i}$ 和 $\overline{k_j}$ 之间的夹角; $\varphi_k, n_{k_i}, n_{k_j}$ 则分别为细节点之间的跨越脊线数.

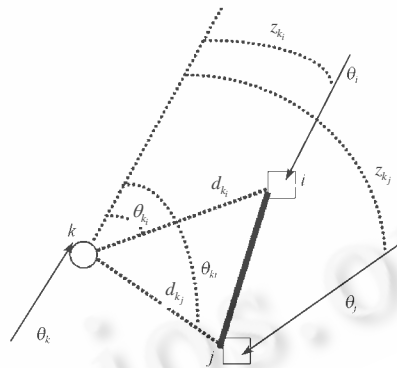


Fig.6 Structure of minutia pair for matching

图6 待比对的点对结构

3.以 Fl'_k 为模板, Fl_k 为待匹配指纹所提取的模板.在匹配过程中,利用两个矢量分别记录匹配过程信息.分别求出两个矢量之间的平移和旋转参数.

4.在求得平移和旋转参数之后,将待匹配指纹模板矢量 Fl_k 通过反向平移和旋转,与模板矢量 Fl'_k 配准.在第2次匹配中,由于两个点集矢量都已在统一的坐标系中配准,因此只需比较细节点之间的位置差异,通过罗希平等人^[12]提出的弹性限界盒方法计算出匹配分数.

3 实验结果

为了验证本文提出的快速算法的分类和匹配性能,我们从两方面对算法在大数据库上的整体表现进行评估,以NIST DB4和FVC2004 DB1作为算法测试数据库,设计分类实验和匹配实验,分别用来测试算法的分类效率和正确率,以及匹配速度和正确率.这样可以正确地反映快速识别算法全方面的性能.

NIST DB4 指纹数据库是一个专门测试指纹分类算法性能的数据库,大多数指纹连续分类算法都采用该数据库进行测试.它包含了从 2 000 枚手指采集的 4 000 幅卡片指纹图像,图像大小为 480×512 像素.FVC2004 DB1 作为 FVC2004 竞赛的标准数据库,共有 800 幅指纹,使用 CrossMatch 公司生产的 V300 型光学指纹采集仪采集 100 个不同的手指图像,每个手指采集 8 幅图像.每幅图像的大小为 640×480 像素,分辨率为 500dpi.

3.1 连续分类实验

指纹分类算法性能的评价标准包括两个主要指标:分类正确率和分类效率^[1].分类正确率是用分类正确的指纹图像数量和指纹库中的指纹图像数量的比率来表示.分类效率用分类检索范围来衡量,它是通过分类算法选择出的待匹配指纹数量与指纹库中的指纹图像数量的比率来表示的.

在分类算法测试实验中,我们选用 NIST DB4 和 FVC2004 DB1 数据库分别进行测试.NIST DB4 数据库根据传统指纹分类方式将指纹分成 5 类,每一类 400 枚手指,每一类指纹的分布概率是相同的.为了测试分类算法在指纹自然分布概率下的性能,我们根据指纹自然分布概率从 NIST DB4 指纹库的 5 类指纹中随机选取,得到一个包含 1 184 枚手指的 NIST 测试库(左旋型 400 枚、右旋型 375 枚、漩涡型 330 枚、拱形 44 枚和尖拱形 35 枚).在实验中,将测试库中 S 组指纹图像作为训练集,生成指纹模板库,以 F 组指纹图像为测试集,作为输入指纹.

FVC2004 DB1 数据库中,每个手指采集 8 幅不同情况下的指纹图像,编号分别是 1~8.我们将不同手指的相同编号图像归为同一组,共构成 8 组,每组 100 枚手指的 FVC 测试库.测试过程中,依次选择一组指纹图像作为训练集,生成指纹模板库,以其他 7 组指纹图像作为测试集,分别进行测试,共进行 28 组测试.最终的分类测试结果分别为各组测试结果的平均值.

图 7 为本文提出的奇异点邻近结构分别在 NIST 测试库和 FVC 测试库上的分类检索性能曲线.从中可以看到,在检索范围为 11% 时,算法在 NIST 测试库和 FVC 测试库上的分类正确率相同,都为 89.5%.在检索范围大于 11% 时,算法在 NIST 测试库上的分类检索性能比在 FVC 测试库上的要好,同时,随着检索范围的增大,算法在 FVC 测试库上的分类正确率没有明显提高.这主要是因为 FVC 测试库的指纹图像在采集过程中指纹存在着较大的形变和位移,部分指纹的 Core 点已经位于图像边界或不在指纹图像有效区域中.经过人工统计,出现这种情况的图像占整个 FVC 测试库的 3%.

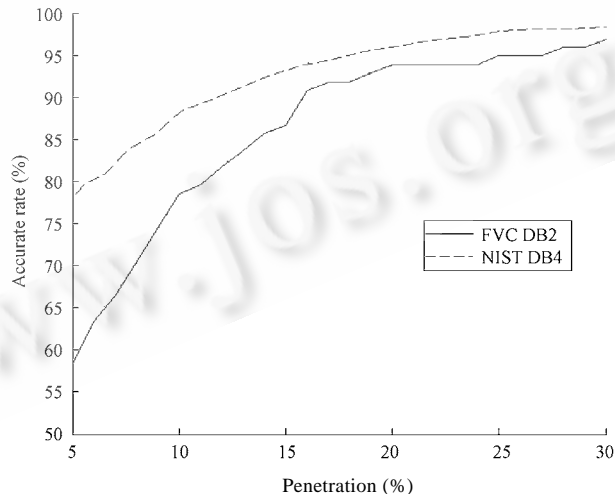


Fig.7 Experimental result of our algorithm on NIST DB4

图 7 本文的指纹分类算法在 NIST DB4 数据库上的分类效果

与此同时,我们将本文的分类算法与Bhanu^[7]和Li^[8]的算法分别进行了比较.其中,Bhanu算法中所用的分类特征是细节点三角结构中定义的一些特征,由于细节点属于局部特征,细节点按照一定规则组成的三角结构具

有很强的抗形变能力,因此对于形变较大的数据库,Bhanu算法所计算的分类特征具有较高的稳定性,分类效果较好.而Li算法是基于对称性测量的指纹连续分类算法,通过计算不同方向特征的滤波相应地进行分类.实验结果表明,Li算法有较高的分类效率.表 1 中的数据为各分类算法在不同检索范围下的分类正确率.本文提出的分类算法在检索范围小于 10%时,性能不如Bhanu和Li的算法,但是 3 种分类算法的正确率都小于 90%.在检索范围大于 10%的情况下,本文提出的分类算法的分类正确率要高于上述两种算法,而且随着检索范围增加到 19%,分类正确率上升得很快,已经达到 95.6%.

Table 1 Accurate rates of some continuous classification algorithms and our algorithms on NIST DB4

表 1 本文算法与其他几种连续分类算法在 NIST DB4 上的分类正确率的比较

Percent of database	7%	10%	13%	16%	19%	22%
Bhanu's ^[7] algorithm	85.2%	88.1%	89.9%	91.4%	92.35%	-
Li's ^[8] algorithm	86.0%	88.2%	91.0%	93.5%	95.15%	96.2%
Algorithm in this paper	82.5%	88.4%	91.4%	94.0%	95.6%	96.9%

我们同时将本文的算法与最新技术水平的固定类别分类算法进行比较.大多数已发表的固定类别分类算法以 NIST DB4 作为测试库,将指纹划分为 4 类或 5 类.根据文献[4]中的平均检索范围计算方法,对于 NIST DB4 数据库,将其划分为 5 类时的平均检索范围为 20%,将其划分为 4 类时的平均检索范围为 28%(此时,将拱形和尖拱形指纹划分为同一类).同时,一些学者为了能够使用 NIST DB4 测试自然指纹分布情况下的固定类别指纹分类算法性能,根据自然指纹分布比例提取一个新的测试库或者将分类结果进行加权处理.在分类结果加权处理的情况下,将数据库划分为 5 类时的平均检索范围为 29.48%,将其划分为 4 类时的平均检索范围为 29.69%.表 2 显示了本文提出的算法与 4 种已发表的固定类别分类算法在 NIST DB4 上的比较.表 2 中的数据为各分类算法在不同条件下的分类正确率.

Table 2 Accurate rates of some exclusive classification algorithms and our algorithm on NIST DB4

表 2 本算法与几种固定分类算法在 NIST DB4 上的分类正确率的比较

Indexing algorithm	5-Class classifier searching range=20%	Weighted 5-Class classifier searching range=28%	4-Class classifier searching range =29.48%	Weighted 4-Class classifier searching range=29.69%
Zhang's ^[18] algorithm	84.3%	-	92.7%	-
Yao's ^[19] algorithm	89.3%	91.0%	93.1%	-
Park's ^[20] algorithm	90.7%	-	94.0%	-
Cappelli's ^[21] algorithm	92.1%	93.5%	94.5%	-
Algorithm in this paper	96.1%	98.3%	98.1%	98.3%

3.2 快速匹配实验

由于本文的快速算法将奇异点邻近结构代替整幅指纹图像进行匹配特征提取,使得一些细节点特征未被提取到模板之中.从全局角度来讲,这样会使比对信息部分缺失,从而造成匹配性能在一定程度上的下降.因此,本文设计了在 FVC2004 数据库上的匹配实验,用于测试本算法的匹配速度,同时验证算法的识别性能,并与 FVC2004 官方的测试结果相比较,证明本快速算法虽然在提高识别速度的同时造成了识别正确率上的一些损失,但是整体的识别性能仍然保持在一个很高的水平.并且由于 FVC2004 是活体指纹采集的图像,因此在 FVC2004 数据库上的匹配结果也证明了算法在指纹实时在线比对中的性能.

根据 FVC2004 的测试规则,对每个手指的所有 8 幅图像进行两两之间匹配,而对所有手指的第 1 幅图像进行两两之间匹配,这样对整个数据库进行了 $100 \times (8 \times 7) / 2 = 2800$ 次真匹配,以及 $(100 \times 99) / 2 = 4950$ 次假匹配.

首先,我们用使用了奇异点邻近结构的算法和未使用邻近结构的点匹配算法进行了对比实验,用于测试本算法在识别准确率上的表现.实验结果以算法运行曲线(receiver operating characteristic curves,简称 ROC 曲线)来表示,浅色曲线是未采用邻近结构比对算法的 ROC 曲线,深色曲线为采用邻近结构算法的 ROC 曲线,如图 8 所示.

实验表明,在应用了奇异点邻近结构的算法中,DB1 的平均模板大小为 0.8KB.实验中,我们采用了主频为 2.8GHz 的 P4 CPU,512MB RAM 内存,以及 Window XP 系统的计算机环境.实验结果表明,在 FVC2004 DB1 上

的平均匹配时间为 0.001 秒,比普通算法的平均匹配时间大为降低.为了进一步说明本算法的优越性,通过与我们的改进的 FVC2004 Light 类参赛算法的结果相比较,可以得出表 3 的结果.在表 3 中,我们对两种算法的平均等错误率(equal error rate,简称 EER)、平均注册和匹配时间以及平均模板大小进行全面的比较.EER 反映了指纹识别算法的识别率,平均运行时间和平均模板大小则反映了算法对资源的消耗和执行速度,这些参数反映了算法在大容量数据库环境下的运行效率,也是我们进行快速指纹识别算法研究的主要目标.

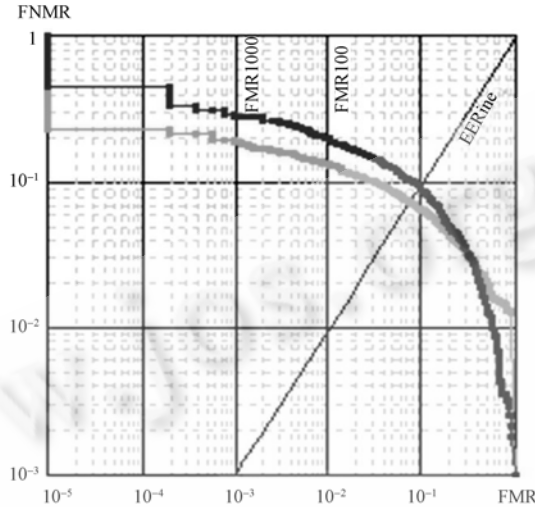


Fig.8 ROC-Curves of experiments on DB1 in FVC2004

图 8 FVC2004 DB1 上的实验的 ROC 曲线

Table 3 Comparison of our algorithm with neighborhood structure and without the structure, each time consuming is supposed on CPU of 2.8GHz

表 3 采用邻近结构和未用邻近结构的本文算法性能比较,在 2.8GHz 的 CPU 上的时间消耗

Algorithm	EER (%)	Mean enroll time (s)	Mean matching time (s)	Mean template size (KB)
Algorithm without neighborhood structure	7.31	0.221	0.003	1.0
Algorithm with neighborhood structure	9.25	0.031	0.001	0.8

其中,平均注册时间是包含了图像增强部分的时间,而平均匹配时间是指纹模板对模板进行比对的时间,不包含图像增强部分.在 FVC2004 竞赛 DB1 数据库中,Light 类 EER 排名第 10 的算法 EER 为 9.92%.表 3 的结果表明,采用邻近结构后的匹配算法在识别性能上略有损失,但是仍然能够在 FVC2004 竞赛中排名前 10,与此同时,匹配算法的执行速度与未采用邻近结构的算法相比大为提高,并且算法的平均模板大小也有所降低.以上结果说明,采用基于奇异点邻近结构的快速指纹识别算法在保证识别率的同时,在资源消耗和速度等方面比传统 Light 类算法有了很大的提高.

4 总结与讨论

大容量指纹数据库上的在线比对给指纹识别算法提出了更高的要求.指纹识别算法不仅要保证识别的准确率,更要提高识别速度以适应实用化的需求.在自动指纹识别系统的算法设计过程中,必须将资源和时间的消耗作为影响性能的重要因素考虑进去.本文提出了一种新的奇异点邻近结构,利用该结构可以进行连续指纹分类和快速指纹匹配,将两种加速手段统一于快速指纹识别的算法之中.实验结果表明,该算法在分类和匹配过程中均表现良好.在两个数据库上的分类实验结果表明,在检索范围较大的大容量指纹数据库中,本算法的分类正确率达到了 90% 以上.FVC2004 DB1 上的匹配实验结果也说明了本算法在提高系统运行速度的同时,识别准确

率仅有极小的降低,同时,该实验结果更加证明了该算法能够有效地降低系统资源和时间消耗,具有很高的运行速度和较高的识别精度,能够符合大容量指纹数据库在线识别的需要。

与此同时,目前的算法在加速手段上还有提高的空间。首先,在结构中的细节点配对策略还可以加以改进,通过对细节点建立索引的方法,进一步减少细节点对之间的匹配计算时间;其次,可以在分类和匹配时对奇异点邻近结构中的同心圆半径分别进行调整,以分别适应于分类和匹配的不同过程,提高分类和匹配的正确率,进而提高算法的整体性能。在进一步的研究工作中,将对本算法的连续分类和快速匹配算法加以改进和优化,提高算法的正确率和执行效率,使其能够更加适应于大容量指纹数据库在线比对的需求。

References:

- [1] Maltoni D, Maio D, Jain AK, Prabhakar S. *Handbook of Fingerprint Recognition*. New York: Springer-Verlag, 2003. 173–202.
- [2] Chang JH, Fan KC. A new model for fingerprint classification by ridge distribution sequences. *Pattern Recognition*, 2002,35(6): 1209–1223.
- [3] Tan XJ, Bhanu B, Lin YQ. Fingerprint classification based on learned features. *IEEE Trans. on Systems, Man, and Cybernetics*, 2005,35(3):287–300.
- [4] Lumini A, Maio D, Maltoni D. Continuous versus exclusive classification for fingerprint retrieval. *Pattern Recognition Letters*, 1997,18(10):1027–1034.
- [5] Cappelli R, Maio D, Maltoni D. Combining fingerprint classifiers. In: Josef K, Fabio R, eds. *Proc. of the Multiple Classifier Systems*. Berlin: Springer-Verlag, 2000. 351–361.
- [6] Boer JD, Bazen AM, Gerez SH. Indexing fingerprint database based on multiple features. In: *Proc. of the ProRISC Workshop on Circuits, Systems and Signal Processing*. 2001. <http://utelnt.el.utwente.nl/links/gerez/publications/pslist.html>
- [7] Bhanu B, Tan XJ. Fingerprint indexing based on novel features of minutiae triplets. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2003,25(5):616–622.
- [8] Li J, Wang H. Fingerprint indexing based on symmetrical measurement. In: Tang YY, Wang SP, Lorette G, Yeung DS, Yan H, eds. *Proc. of the 18th Int'l Conf. on Pattern Recognition*. Washington: IEEE Computer Society, 2006. 1038–1041.
- [9] NIST Special Database 4, Fingerprint Database. 1992. <http://www.nist.gov/srd/nistsd4.htm>
- [10] Fingerprint verification competition. 2004. <http://bias.csr.unibo.it/fvc2004/>
- [11] Hong L, Wan YF, Jain AK. Fingerprint image enhancement: Algorithm and performance evaluation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1998,20(8):777–789.
- [12] Luo XP, Tian J. Image enhancement and minutia matching algorithms in automated fingerprint identification system. *Journal of Software*, 2002,13(5):946–956 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/13/946.pdf>
- [13] Kawagoe M, Tojo A. Fingerprint pattern classification. *Pattern Recognition*, 1984,17(3):295–303.
- [14] Liu MH, Jiang XD, Kot AC. Reference point detection for fingerprint recognition. In: Zhang D, Jian AK, eds. *Biometric Authentication*. Berlin: Springer-Verlag, 2004. 272–279.
- [15] Lee D, Choi K, Kim J. A robust fingerprint matching algorithm using local alignment. In: Kasturi R, Laurendeau D, Suen C, eds. *Proc. of the 16th Int'l Conf. on Pattern Recognition*, Vol.3. Washington: IEEE Computer Society, 2002. 803–806.
- [16] Chen H, Tian J. A fingerprint matching algorithm with registration pattern inspection. *Journal of Software*, 2005,16(6):1046–1053 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/16/1046.htm>
- [17] Jiang XD, Yau WY. Fingerprint minutiae matching based on the local and global structures, In: Sanfeliu A, Villanueva JJ, Vanrell M, Alquezar R, Crowley J, Shirai Y. eds. *Proc. of the 15th Int'l Conf. on Pattern Recognition*. Washington: IEEE Computer Society, 2000. 1038–1041.
- [18] Zhang Q, Yan H. Fingerprint classification based on extraction and analysis of singularities and pseudo ridges. *Pattern Recognition*, 2004,37(11):2233–2243.
- [19] Yao Y, Frasconi P, Pontil M. Fingerprint classification with combination of support vector machines. In: Josef B, Fabrizio S, eds. *Audio- and Video-Based Biometric Person Authentication*. Berlin: Springer-Verlag, 2001. 253–25.
- [20] Park CH, Park H. Fingerprint classification using fast fourier transform and nonlinear discriminant analysis. *Pattern Recognition*, 2005,38(4):495–503.

