

自动指纹识别中的图像增强和细节匹配算法*

罗希平, 田捷

(中国科学院 自动化研究所 人工智能实验室,北京 100080)

E-mail: tian@dr.com

http://www.fingerpass.net

摘要: 对自动指纹识别系统(automated fingerprint identification system,简称 AFIS)的两个重要问题——指纹图像增强和细节匹配进行研究,给出了一种基于方向场的指纹图像增强算法,对 Anil Jain 等人提出的细节匹配算法进行了修正.采用一种新的更简单的方法进行指纹图像的校准,并以一种简单而有效的方式将脊线信息引入匹配过程中,这样做的好处之一是以较低的计算代价有效地解决了匹配中参照点对的选取问题.另外,采用大小可变的限界盒来适应指纹的非线性形变.在 FVC2000 公布的指纹图像数据库上,按照 FVC2000 测试标准所做的实验显示,该算法比原算法有较大的改进.

关键词: 自动指纹识别系统(AFIS);图像增强;细节匹配

中图法分类号: TP391 文献标识码: A

1 背景介绍

自动指纹识别系统(automated fingerprint identification system,简称 AFIS)有着广泛的应用背景.目前对自动指纹识别系统的研究主要有 3 个方面,即图像增强、指纹分类和细节匹配.指纹分类一般是用在大规模的指纹库中,作为细节匹配中减少搜索范围的步骤使用.本文的工作将不涉及指纹分类.

我们的自动指纹识别系统框图如图 1 所示,系统由离线和在线部分两部分组成.在系统的离线部分,用指纹采集仪采集指纹,提取出细节点,然后将细节点保存到数据库中,形成指纹模板库.在系统的在线部分,用指纹采集仪采集指纹,提取出细节点,然后将这些细节点与保存在数据库中模板细节点进行匹配,判断输入细节点与模板细节点是否来自同一个手指的指纹.

在实践中,由于受手指本身的因素和采集条件的影响,采集到的指纹图像会不同程度地受到各种噪声的干扰,在进行分类和细节匹配之前一般要对采集到的指纹图像做增强处理.指纹图像增强算法多数是基于方向场估计的图像滤波算法^[1-3],我们采用的也是一种基于方向场估计的图像滤波算法.

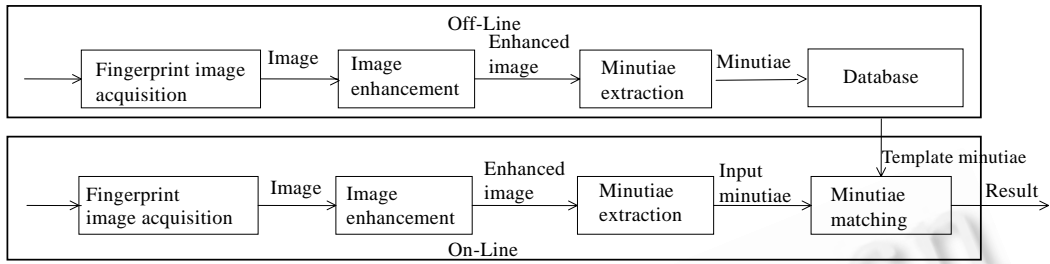
指纹匹配是指纹识别系统的核心步骤,也是一个重要的研究课题,人们在这方面作了很多工作,D.K.Isenor 等人^[4]提出了一种用图匹配来对两幅指纹图像进行匹配的方法.Andrew K.Hrechak 等人^[5]用结构匹配来做指纹识别.但目前最常用的方法是用 FBI 提出的细节点坐标模型来做细节匹配.它利用脊线末梢(我们称为端点)与脊线分支点这两种关键点来鉴定指纹.通过将细节点表示为点模式,一个自动指纹认证问题可以转化为一个点模式匹配(细节匹配)问题.一般的点模式匹配问题是模式识别中的一个有名的难题,人们对一般的点模式匹配问题提出过很多的算法,像 Sanjay Ranade 等人^[6]的松弛算法、Shih-hsu Chang 等人^[7]的基于二维聚类的快速算法、Zsolt Miklós 等人^[8]的三角匹配的算法、Xudong Jiang 等人^[9]的基于局部和全局结构的匹配算法.Anil Jain 等

* 收稿日期: 2001-04-06; 修改日期: 2001-07-06

基金项目: 国家自然科学基金资助项目(69875019;60071002;60072007;69931010;60172057)

作者简介: 罗希平(1974 -),男,湖南娄底人,博士,主要研究领域为图像处理,模式识别;田捷(1960 -),男,安徽芜湖人,博士,研究员,博士生导师,主要研究领域为医学图像处理,模式识别,计算机图形学.

人^[3]在针对指纹匹配中的点模式匹配问题提出了一种算法,该算法将直角坐标系中的细节点转换到极坐标系中,通过串匹配算法来进行点匹配。



离线部分, 指纹图像采集, 指纹图像, 图像增强, 增强后图像, 细节提取, 细节点, 细节点数据库, 输入细节点, 细节匹配, 模板细节点, 匹配结果, 在线部分。

Fig.1 Block diagram of AFIS

图 1 自动指纹识别系统框图

我们参考 Anil Jain 等人^[3]的算法,并对其作较大的修改,提出一种新的基于极坐标下的点模式匹配的指纹匹配算法.修改将主要体现在 3 个方面.首先,我们采用一种更为简单而有效的指纹图像校准方法.其次,与 Anil Jain 等人的算法仅在校准阶段使用脊线信息的做法不同,我们将脊线信息引入随后的匹配过程中.最后,Anil Jain 等人的算法在匹配过程中使用了一个固定大小的界定盒,而我们的算法将采用一个大小可变的界定盒,从而使算法能更有效地处理被匹配的两幅指纹图像之间的非线性形变,被匹配的两幅指纹图像之间的非线性形变是指纹匹配中最难解决的问题,是对指纹匹配算法最大的挑战.本文的前期工作见文献[10].

本文第 2 节介绍指纹图像增强算法.第 3 节介绍细节提取.第 4 节介绍细节匹配算法.第 5 节给出实验结果.第 6 节是总结和相关的讨论.

2 指纹图像增强

2.1 方向场估计

一般的指纹图像都有比较清晰的方向场,方向场估计的准确性直接决定了图像增强算法的效果.为估计方向场,我们把指纹脊线的走向分为 8 个方向,如图 2 所示.

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

Fig.2 Eight ridge directions of a pixel

图 2 在一个像素处的 8 个指纹脊线方向

对于图像的每一个像素,为了确定在该像素处脊线的方向,在以该像素为中心的 9×9 窗口内,分别计算 8 个方向上的灰度平均值,即将图 2 中标有 $i(i=0,1,\dots,7)$ 分别代表 8 个方向)的位置的像素灰度值取平均得到 $Gmean[i]$,然后将这 8 个平均值按两两垂直的方向分成 4 组,0 和 4 一组,1 和 5 一组,2 和 6 一组,3 和 7 一组,计算每组中两个平均值的差值

$$Gdiff[j]=abs(Gmean[j]-Gmean[j+4]), j=0,1,2,3 \text{ 为脊线方向,}$$

取差值的绝对值最大的两个方向为可能的脊线方向,若

$$iMax = \arg(\text{Max}_i(Gdiff(i)),$$

则方向 $iMax$ 和 $iMax+4$ 为该像素处可能的脊线方向,若该像素处的灰度值为 Gray,则脊线方向为

$$iDir = \begin{cases} iMax & \text{if } abs(Gray - Gmean[iMax]) < abs(Gray - Gmean[iMax + 4]) \\ iMax + 4 & \text{otherwise} \end{cases},$$

即取 $iMax$ 和 $iMax+4$ 方向中灰度平均值与该像素的灰度值比较接近的方向作为该像素处的脊线方向。

为了减少噪声的影响,保证方向场估计的准确性,我们采用通用的做法,将图像分成 16×16 的小块,以小块内所有像素的方向的平均值作为该小块的脊线方向,该小块内每个像素的脊线方向都取为该小块的脊线方向,具体做法为,先计算该小块内方向估计为 $i(i=0,1,\dots,7)$ 分别代表 8 个方向的像素个数 $Num[i]$,然后将该小块图像的脊线方向取为

$$Dir = \arg(\text{Max}_i(\text{Num}[i])).$$

2.2 二值化

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 $iVar=(i+4) \bmod 8$ 的灰度平均值 $Gmean[i]$ 和 $Gmean[iVar]$,然后将该像素二值化为

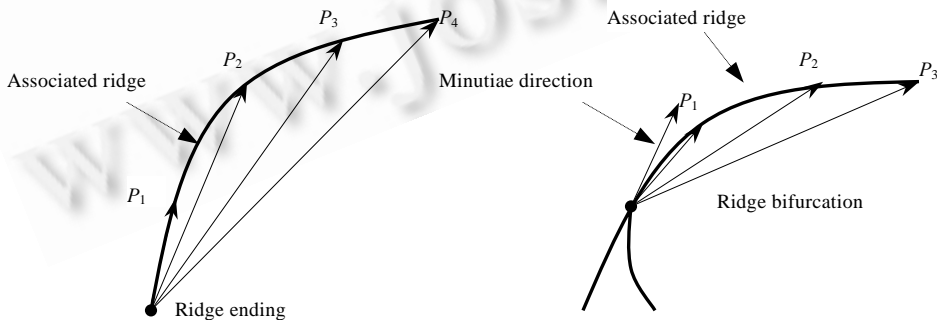
$$iVal = \begin{cases} 255 & \text{if } Gmean[i] \geq Gmean[iVar] \\ 0 & \text{otherwise} \end{cases},$$

其中 $iVar$ 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。

3 细节提取

经过上述图像增强步骤之后,再对二值图像进行细化,检测细节点及进行细节点后处理,在此不详叙。对最终检测出来的每一个细节点,我们记录如下信息:(1) 细节点的 x,y 坐标;(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,即脊线末梢或脊线分支;(4) 细节点对应的脊线 (d_i, α_i) 。这样就将一副指纹图像转化成了一个由细节点组成的平面点集。

细节点对应的脊线用在与该脊线上的采样点来表示,采样的距离约为脊线间的平均距离。脊线分支点对应的脊线是距该细节点的方向最近的那条。脊线末梢对应的脊线则是该细节点所在的脊线。采样点用该点与对应细节点的距离 d_i 和连接该点与对应细节点的直线与对应细节点方向的夹角 α_i 来表示, α_i 的取值范围是 $-180^\circ \sim 180^\circ$ 。图 3 给出了细节点对应的脊线及脊线上的采样点的例子。在细节匹配中,对应脊线将被用来对需要进行匹配的两个平面点集进行校准,而且校准的参数,也就是两个点集中任意一对脊线间的旋转角度,将被用作判断它们所对应的细节点能否看做是匹配的细节点的条件。



对应脊线, 细节点方向, 脊线末梢, 脊线分支。

Fig.3 Associated ridge

图 3 细节点的对应脊线

4 细节匹配

4.1 细节点集的校准

令 $P = ((x_1^P, y_1^P, \theta_1^P)^T, \dots, (x_M^P, y_M^P, \theta_M^P)^T)$ 表示模板图像中的 M 个细节点, $Q = ((x_1^Q, y_1^Q, \theta_1^Q)^T, \dots, (x_N^Q, y_N^Q, \theta_N^Q)^T)$ 表示输入图像中的 N 个细节点.

为了把细节点转换到极坐标系中去,要在模板细节点集和输入细节点集中各选一个参照点作为相应的极坐标系中的原点,并算出其他细节点相对于参照点的极坐标.

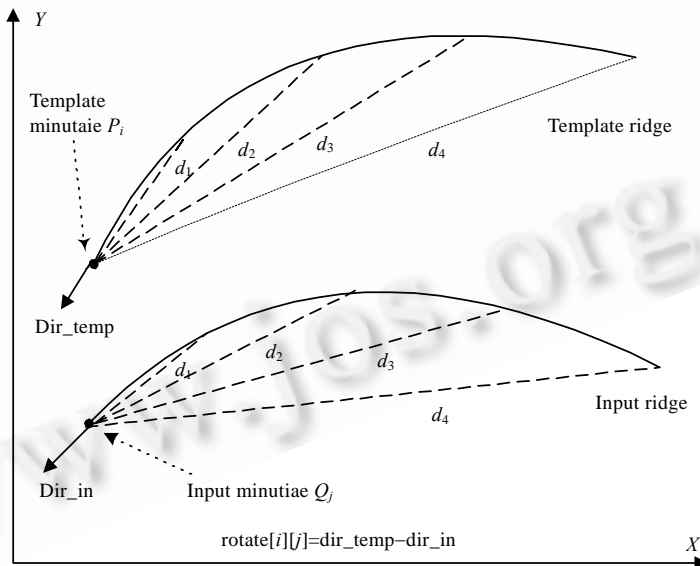
对模板点集中的每一点 $P_i(1 \leq i \leq M)$ 和输入点集中的每一点 $Q_j(1 \leq j \leq N)$,定义 $rotate[i][j]$ 为将 P_i 和 Q_j 当作参照点对时,从输入图像到模板图像的旋转角度.如果 P_i 与 Q_j 可以被当成一对对应点,即它们分别对应的脊线相似性到一定程度,则 $rotate[i][j]$ 将取 $0^\circ \sim 360^\circ$ 间的一个值,否则我们定义 $rotate[i][j]$ 取值为 400 以表示 P_i 与 Q_j 不能为一对对应点.注意, $rotate[i][j] < 400$ 表示 P_i 与 Q_j 对应的脊线相似性到了一定程度.

如果 P_i 和 Q_j 是相同类型的细节点,且记录的对应脊线中的点个数相同,则它们(P_i 与 Q_j)是不是对应点对及 $rotate[i][j]$ 的取值将由如下算法决定.

用 R 表示细节点 P_i 对应的脊线, r 表示细节点 Q_j 对应的脊线.匹配 r 与 R ,用式(1)计算这两条脊线间的差异:

$$\left. \begin{aligned} Diff_dist &= \frac{1}{L} \sum_{i=0}^L |R(d_i) - r(d_i)|, \\ Diff_ang &= \frac{1}{L} \sum_{i=0}^L |R(\alpha_i) - r(\alpha_i)|, \end{aligned} \right\} \quad (1)$$

其中 L 是记录的脊线中的点个数, $R(d_i)$ 和 $r(d_i)$ 分别表示从脊线 R 与 r 上的点 i 到对应的细节点的距离, $R(\alpha_i)$ 和 $r(\alpha_i)$ 分别表示连接脊线 R 与 r 上的点 i 与对应的细节点的直线与对应细节点的方向的夹角,如图 4 所示.



模板细节点 P_i , 模板脊线, 输入细节点 Q_j , 输入脊线.

Fig.4 Alignment of the input ridge and the template ridge

图 4 输入脊线与模板脊线的校准

如果这两条脊线的差异 $Diff_dist$ 和 $Diff_ang$ 分别小于某个阈值 T_d 和 T_a ,也就是说这两条脊线的形状在一定程度上相似,那么 P_i 和 Q_j 能被当作对应细节点对, $rotate[i][j]$ 为

$$rotate[i][j] = dir_temp - dir_in, \quad (2)$$

其中 dir_temp 和 dir_in 分别是 P_i 和 Q_j 的方向.

如果 $Diff_dist$ 大于 T_d 或 $Diff_ang$ 大于 T_a , 也就是说两条脊线彼此不相似, 那么细节点 P_i 与 Q_j 不能被当作对应细节点对, 我们将 $rotate[i][j]$ 的值设为 400.

由于最后要将细节点都转换到极坐标系下, 我们仅仅计算了输入图像与模板图像间的旋转角度, 而不考虑两幅图像间的平移, 也不会像文献[4]中那样真的把 N 个输入细节点做坐标旋转运算. 要在极坐标系中将输入图像与模板图像校准, 只需将输入细节点与模板细节点都分别相对于参照点 P_i 和 Q_j 转换到极坐标系中, 然后在所有输入细节点的极角上加一个角度 $rotate[i][j]$. 也就是说, 将输入细节点与模板细节点都分别相对于参照点 P_i 和 Q_j 用下式转换到极坐标系中

$$\begin{pmatrix} r_i \\ e_i \\ \theta_i \end{pmatrix} = \begin{pmatrix} \sqrt{(x_i - x^r)^2 + (y_i - y^r)^2} \\ \tan^{-1} \left(\frac{y_i - y^r}{x_i - x^r} \right) \\ \theta_i - \theta^r \end{pmatrix}, \tag{3}$$

其中 $(x_i, y_i, \theta_i)^T$ 是待转换细节点的坐标, $(x^r, y^r, \theta^r)^T$ 是参照细节点的坐标, $(r_i, e_i, \theta_i)^T$ 是细节点在极坐标系中的表示 (r_i 表示极半径, e_i 表示极角, θ_i 表示细节点相对于参照细节点的方向). 然后, 我们对每一个输入细节点的 e_i 加一个角度 $rotate[i][j]$.

4.2 校准后的细节匹配

我们的细节匹配算法步骤如下:

- (1) 对每一个 $i(1 \leq i \leq M)$ 和每一个 $j(1 \leq j \leq N)$, 如果 $rotate[i][j]=400$, 即细节点 P_i 与 Q_j 不能被当作对应细节点对, 则重复此步并选择另一对 P_i 和 Q_j , 否则转向步骤(2). 如果所有的细节点对都已考虑过了, 则转向步骤(5).
- (2) 将 P_i 和 Q_j 当作参照细节点, 用第 4.1 节中的方法将输入点集和模板点集中的细节点都转换成极坐标.
- (3) 将极坐标中的模板细节点和输入细节点按极角递增方向排序, 并连接成串, 表示如下:

$$P_i^s = \left((r_1^P, e_1^P, \theta_1^P)^T, \dots, (r_M^P, e_M^P, \theta_M^P)^T \right), \tag{4}$$

$$Q_j^s = \left((r_1^Q, e_1^Q, \theta_1^Q)^T, \dots, (r_N^Q, e_N^Q, \theta_N^Q)^T \right), \tag{5}$$

其中 $(r_s^P, e_s^P, \theta_s^P)$ 和 $(r_s^Q, e_s^Q, \theta_s^Q)$ 表示对应的极半径、极角和相对于参照点的细节点方向.

- (4) 用后面将要介绍的方法匹配串 P_i^s 和 Q_j^s , 找出匹配分数, 记录为 $m_score[i][j]$. 然后转回步骤(1).
- (5) 找出 $m_score[i][j]$ 中的最大值, 把它当作输入细节点集与模板细节点集的匹配分数. 如果匹配分数高于一个预先设定的阈值, 则认为输入图像与模板图像来自同一个指纹, 否则认为它们来自不同指纹.

在介绍匹配串 P_i^s 和 Q_j^s 的方法之前, 我们先介绍一下限界盒及其大小. 如图 5 所示, 一个限界盒是放在模板

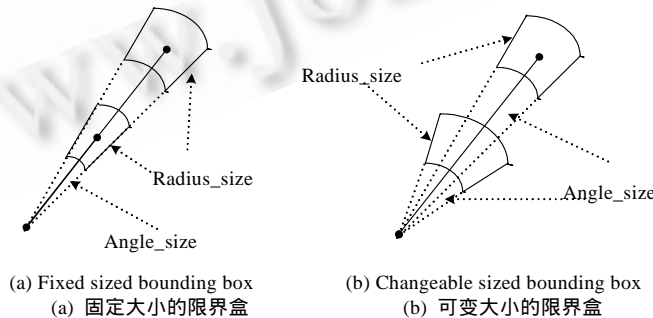


Fig.5

图 5

细节点上的一个盒子. 限界盒的大小用 $angle_size$ 和 $radius_size$ 来表示.

文献[3]中使用了一个固定大小的限界盒, 即在所有的模板细节点处, $angle_size$ 和 $radius_size$ 取同样的值.

我们将使用可变大小的限界盒,即 $angle_size$ 和 $radius_size$ 的值将随着细节节点的极径大小而改变.如果模板细节点的极径比较大,它的限界盒将有一个较大的 $radius_size$ 和较小的 $angle_size$.固定大小与可变大小限界盒的差异可从图 5 中看出来.用下式来计算极半径为 r 的模板细节点的 $radius_size$ 和 $angle_size$.

$$radius_size = \begin{cases} r_small & \text{if } r_size < r_small \\ r_size & \text{if } r_small < r_size < r_large, \\ r_large & \text{if } r_size > r_large \end{cases}, \quad (6)$$

$$r_size = r_small + \frac{r}{\alpha}, \quad (7)$$

$$angle_size = \begin{cases} a_small & \text{if } a_size < a_small \\ a_size & \text{if } a_small < a_size < a_large, \\ a_large & \text{if } a_size > a_large \end{cases}, \quad (8)$$

$$a_size = \frac{r_size}{r}, \quad (9)$$

其中 r 是模板细节点的极径 $r_small, r_large, a_small, a_large$ 分别是 $radius_size$ 和 $angle_size$ 的上界和下界,它们的值是预先设定的. α 是预先给定的常数.

我们使用可变大小的限界盒而不是固定大小的限界盒是为了使算法对非线性形变更为鲁棒.非线性形变一般在一个特定的区域内较大,然后非线性地向外扩张.当细节点的极径较小时,小的形变就可以造成大的极角的改变,而极半径的改变较小.所以在这种情况下,限界盒的 $angle_size$ 应该比较大而 $radius_size$ 则应该比较小.另一方面,当细节点的极半径较大时,极角的较小改变就会造成细节点位置的较大变动,而极半径的形变可以看成是该细节点与参照细节点间的所有区域的形变的累加.所以在这种情况下,限界盒的 $angle_size$ 应该比较小而 $radius_size$ 则应该较大.

匹配 P_i^s 和 Q_j^s 的算法描述如下:

(1) 用式(6)~式(9)决定每一个模板细节点的限界盒的大小.置 $m_score[i][j]=0$.

(2) 进行如下循环:

While $1 \leq k \leq M$ do

While $1 \leq L \leq N$ and $e_L^P < angle_high[k]$ do

if $template_point[k]$ and $input_point[L]$ satisfy condition1, then

$m_score[i][j] = m_score[i][j] + 1$;

调整限界盒;

end if

Increase L ;

End while

Increase k ;

End while

上述过程中,condition1 定义为

$$condition1 = \begin{cases} \text{true} & \text{if } \begin{cases} \text{在限界盒内} \\ \Delta\theta < \varepsilon \\ rotate[k][L] < 400 \end{cases} \\ \text{false} & \text{otherwise} \end{cases}, \quad (10)$$

$$\Delta e = \begin{cases} a & \text{if } (a = (e_L^P - e_k^Q + 360) \bmod 360) < 180 \\ a - 180 & \text{otherwise} \end{cases}, \quad (11)$$

$$\Delta e = \begin{cases} a & \text{if } (a = (\theta_L^P - \theta_k^Q + 360) \bmod 360) < 180 \\ a - 180 & \text{otherwise} \end{cases}. \quad (12)$$

Condition1 是将 $template_point[k]$ 和 $input_point[L]$ 看作对应点对的条件.其含义是, $input_point[L]$ 应该在 $template_point[k]$ 的限界盒的内部,这两个细节点的方向差异应小于 ϵ (我们设置 $\epsilon=30$), $rotate[k][l]$ 应小于 400,即 P_k 和 Q_l 对应的脊线不相似.

将体现在 $rotate[i][j]$ 中的脊线信息引入到匹配过程中来的作用主要是两个方面.一方面是在指纹细节匹配中,如何选择可靠的参照点对一直是一个难题,而如果将所有可能的点对分别当作参照点对来计算又将带来计算量太大的问题.我们的算法中通过引入脊线信息,在参照点对的选取中,细节点 P_i 与 Q_j 能被当作参照点对,则必须 $rotate[i][j]<400$,即 P_i 与 Q_j 对应的脊线有一定程度的相似性,这样做大大减少了可能的参照点对的数目,使得我们的算法既保证了计算的速度,又有效地解决了参照细节点对的选取问题.另一方面是引入脊线信息能有效地防止两幅来自不同指纹的图像被错误地匹配为来自同一个指纹的,同时,它对来自同一幅指纹的图像的匹配的影响很小.这是因为如果两幅图像来自同一个指纹,则它们的对应点所对应的脊线往往也是相似的,而如果两幅图像来自不同指纹,则很可能点对 P_k 和 Q_l 满足作为对应点对的所有其他条件,但 $rotate[k][l]=400$,即它们对应的脊线不相似.

5 实验结果

我们在 FVC2000^[11] 公布的 4 个指纹数据库上,按照 FVC2000 的测试标准^[11]对改进后的算法进行了实验,并与改进前的算法作了比较,试验结果见表 1~表 4.

在下面的实验结果报告中,用 Changeable_Ridge 表示使用可变大小的限界盒,并将体现在 $rotate[i][j]$ 中的脊线信息引入到匹配过程中来的匹配算法,用 Fixed_Ridge 表示使用固定大小的限界盒,并在匹配中用到脊线信息的匹配算法,用 Changeable_NoRidge 表示使用可变大小的限界盒并在匹配过程中没有用到脊线信息的匹配算法,用 Fixed_NoRidge 表示使用固定大小的限界盒,并在匹配过程中没有用到脊线信息的匹配算法.这 4 种算法除了存在上述是否使用可变大小的限界盒和是否在匹配过程中用到了脊线信息的差别以外,其他任何条件都是相同的,当使用固定大小的限界盒时,限界盒的大小参数已对每个数据库分别调整到了最佳值.EER,EER*,FNMR,FMR 等指标的定义参见文献[11],在此不再赘述.所有实验均在 PIII450MHz 微机上进行.

由实验结果可以看出,在 FVC2000 的 4 个指纹数据库中,改进后的算法性能均比改进前的算法有明显提高,用可变大小的限界盒代替固定大小的限界盒可以使算法的性能得到明显改善,而在匹配过程中引入脊线信息带来的性能改善相对较小一点,而且在少数特定的情况下可能反而使算法的性能要差一些,如在数据库 DB3 中使用固定大小的限界盒时,不用脊线信息比用脊线信息要好.

5.1 FVC2000数据库DB1

Table 1 Results of the four algorithms on DB1 of FVC2000 Databases

表 1 4 种算法在 FVC2000 数据库 DB1 上的测试结果

Algorithm	EER(%)	EER*(%)	ZeroFMR(%)	ZeroFNMR(%)
Changeable_Ridge	4.39	4.39	21.00	100
Changeable_NoRidge	4.67	4.67	20.11	100
Fixed_Ridge	5.32	5.32	25.03	100
Fixed_NoRidge	5.59	5.59	24.11	100

改进后的算法 Changeable_Ridge 与改进前的算法 Fixed_NoRidge 的 ROC 曲线比较如图 6 所示.

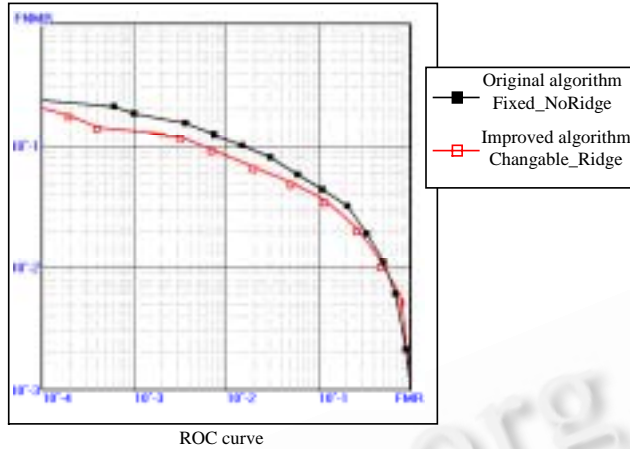


Fig.6 Comparison of the ROC curve of the improved and the original algorithm

图 6 DB1 上改进前、后算法的 ROC 曲线比较

4 种算法对 DB1 的平均注册时间 Avg Enroll Time^[11]和平均匹配时间 Avg Match Time^[11]都是 0.52s.

5.2 FVC2000数据库DB2

Table 2 Result of the four algorithms on DB2 of FVC2000 Databases

表 2 4 种算法在 FVC2000 数据库 DB2 上的测试结果

Algorithm	EER(%)	EER*(%)	ZeroFMR(%)	ZeroFNMR(%)
Changeable_Ridge	3.27	3.27	20.00	100
Changeable_NoRidge	3.50	3.50	21.79	100
Fixed_Ridge	3.94	3.94	18.96	100
Fixed_NoRidge	4.25	4.25	24.46	100

改进后的算法 Changeable_Ridge 与改进前的算法 Fixed_NoRidge 的 ROC 曲线比较如图 7 所示.

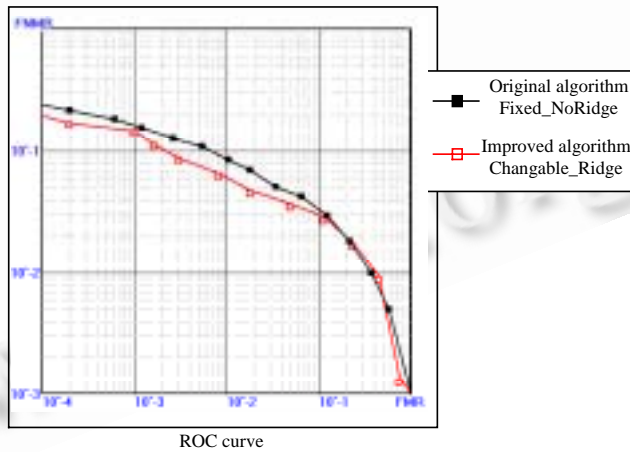


Fig.7 Comparison of the ROC curve of the improved and the original algorithm

图 7 DB2 上改进前、后算法的 ROC 曲线比较

4 种算法对 DB2 的平均注册时间 Avg Enroll Time^[11]和平均匹配时间 Avg Match Time^[11]都是 0.44s.

5.3 FVC2000数据库DB3

Table 3 Results of the four algorithms on DB3 of FVC2000 Databases

表 3 4 种算法在 FVC2000 数据库 DB3 上的测试结果

Algorithm	EER(%)	EER*(%)	ZeroFMR(%)	ZeroFNMR(%)
Changeable_Ridge	10.35	10.35	31.74	100
Changeable_NoRidge	10.71	10.71	36.11	100
Fixed_Ridge	12.37	12.37	39.75	100
Fixed_NoRidge	11.26	11.26	44.64	100

改进后的算法 Changeable_Ridge 与改进前的算法 Fixed_NoRidge 的 ROC 曲线比较如图 8 所示.

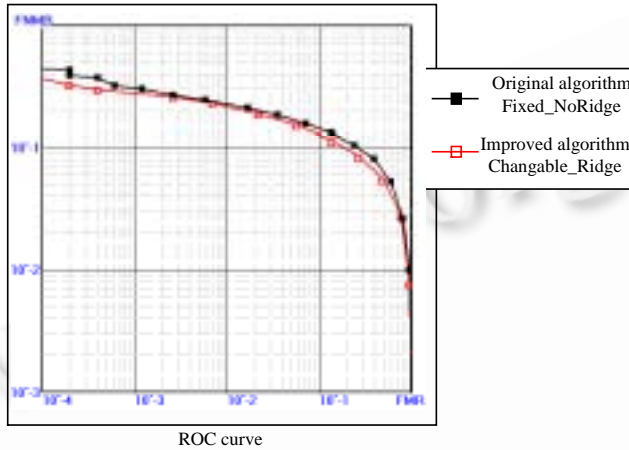


Fig.8 Comparison of the ROC curve of the improved and the original algorithm

图 8 DB3 上改进前、后算法的 ROC 曲线比较

4 种算法对 DB3 的平均注册时间 Avg Enroll Time^[11]和平均匹配时间 Avg Match Time^[11]都是 1.06s.

5.4 FVC2000数据库DB4

Table 4 Results of the four algorithms on DB4 of FVC2000 Databases

表 4 4 种算法在 FVC2000 数据库 DB4 上的测试结果

Algorithm	EER(%)	EER*(%)	ZeroFMR(%)	ZeroFNMR(%)
Changeable_Ridge	6.08	6.08	41.21	100
Changeable_NoRidge	6.47	6.47	39.39	100
Fixed_Ridge	6.61	6.61	43.93	100
Fixed_NoRidge	6.96	6.96	56.71	100

改进后的算法 Changeable_Ridge 与改进前的算法 Fixed_NoRidge 的 ROC 曲线比较如图 9 所示.

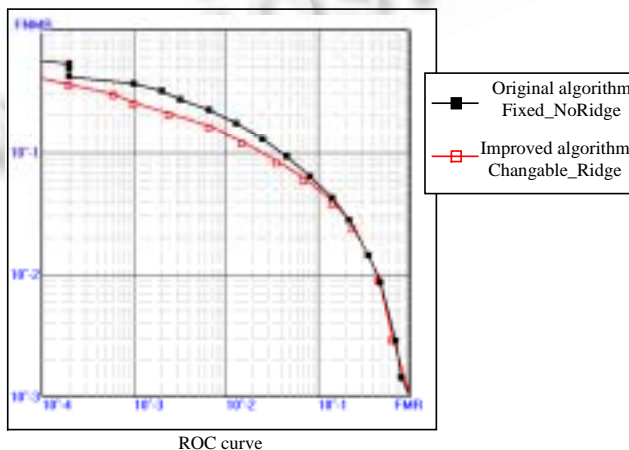


Fig.9 Compare of the ROC curve of the improved and the original algorithm

图 9 DB4 上改进前后算法的 ROC 曲线比较

4种算法对 DB4 的平均注册时间 Avg Enroll Time^[11]和平均匹配时间 Avg Match Time^[11]都是 0.36s.

6 总结与讨论

本文给出了一种基于指纹图像的方向场算法,并主要介绍了一种处理指纹识别中的细节匹配问题的算法.这种细节匹配算法改进了文献[3]中提出的利用极坐标中的点模式匹配来做细节匹配的算法.我们用了一种更简单也更有效的方法来做指纹图像的校准.另外,我们在匹配的过程中引入了脊线信息,并使用了一个大小可变的限界盒,这使得我们的算法具有更强的分辨来自不同指纹的图像的能力,并能更鲁棒地处理指纹图像的非线性形变.

在实验过程中,我们发现大多数指纹图像可以用细节点匹配的方法实现指纹匹配,但也有少数质量很差,脊线结构很不清晰的指纹图像难以用细节点匹配的方法来进行匹配,造成困难的主要原因是,在这样的图像中难以准确地提取出细节点特征来.我们下一步的研究工作将在两个方面展开,一方面是继续改进指纹图像增强算法,更准确地提取细节点,并进一步改进细节匹配算法,增强算法对非线性形变的鲁棒性.另一方面是针对不能准确提取细节点的质量很差的图像研究不是基于细节点匹配的其他指纹匹配算法,作为自动指纹识别系统中细节点匹配方法的辅助方法,考虑多种基于多种特征的匹配方法的结合.

References:

- [1] Huang, D.C. Enhancement and feature purification of fingerprint images. *Pattern Recognition*, 1993,26(11):1661~1671.
- [2] Lin Hong, Wan, Yi-fei, Jain, A. Fingerprint image enhancement: algorithm and performance evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998,20(8):777~789.
- [3] Jain, A., Lin Hong, Bolle, R. On-Line fingerprint verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997,19,(4):302~313.
- [4] Isenor, D.K., Zaky, S.G. Fingerprint identification using graph matching. *Pattern Recognition*, 1986,19(2):113~122.
- [5] Hrechak, A.K., Mchugh, J.A. Automated fingerprint recognition using structural matching. *Pattern Recognition*, 1990,23(8):893~904.
- [6] Ranade, S., Rosenfeld, A. Point pattern matching by relaxation. *Pattern Recognition*, 1980,12(4):269~275.
- [7] Chang, C.H., Cheng, F.H., Hsu, W.H., *et al.* Fast algorithm for point pattern matching: invariant to translations, rotations and scale changes. *Pattern Recognition*, 1997,30(2):311~316.
- [8] Kovács-Vajna, Z.M. A fingerprint verification system based on triangular matching and dynamic time warping. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000,22(11):1266~1276.
- [9] Jiang, Xu-dong, Yau, Wei-Yun. Fingerprint minutiae matching based on the local and global structures. In: Sanfeliu, A., Villanueva, J.J., eds. *Proceedings of the 15th International Conference on Pattern Recognition*. Los Alamitos, CA: IEEE Computer Society Press, 2000. 1042~1045.
- [10] Luo Xi-ping, Tian jie. A minutia matching algorithm in fingerprint verification. In: Sanfeliu, A., Villanueva, J.J. eds. *Proceedings of the 15th International Conference on Pattern Recognition*. Los Alamitos, CA: IEEE Computer Society Press, 2000. 833~836.
- [11] Maio, D., Maltoni, D., Cappelli, R., *et al.* FVC2000: fingerprint verification competition. <http://bias.csr.unibo.it/fvc2000>.

Image Enhancement and Minutia Matching Algorithms in Automated Fingerprint Identification System*

LUO Xi-ping, TIAN Jie

(Artificial Intelligent Laboratory, Institute of Automation, The Chinese Academy of Sciences, Beijing 100080, China)

E-mail: tian@doctor.com

<http://www.fingerpass.net>

Abstract: This paper focuses on two key problems in automated fingerprint identification system (AFIS): fingerprint image enhancement and minutia matching. A fingerprint image enhancement algorithm is proposed based on orientation field. In addition, a minutia matching algorithm is proposed which modified the algorithm presented by Jain et al. About this algorithm, a simpler alignment method is used. Ridge information is introduced into the minutia matching process in a simple but effective way. One of the advantages of doing so is to solve the problem of reference point pair selection with low computational cost. In addition, the changeable sized boundary box is used to make this algorithm more robust to nonlinear deformation between fingerprint images. Experiments are done on the FVC2000 databases with the FVC2000 performance evaluation method and the results show that this improved algorithm is better than the original one.

Key words: automated fingerprint identification system (AFIS); image enhancement; minutia matching

* Received April 6, 2001; accepted July 6, 2001

Supported by the National Natural Science Foundation of China under Grant Nos.69875019, 60071002, 60072007, 69931010, 60172057

第 2 届中国 Rough 集与软计算学术研讨会(CRSSC 2002)

征文通知

由中国计算机学会人工智能与模式识别专业委员会主办、苏州大学承办的第 2 届中国 Rough 集与软计算学术研讨会(CRSSC 2002)拟定于 2002 年 10 月下旬在苏州召开.欢迎高等院校教师、科研院所和企业的科技人员以及博士生、硕士生踊跃参加.

一、征文范围

- (1) Rough 集理论及其应用;Rough 逻辑与 Rough 推理;Rough 代数理论及其应用;
- (2) Fuzzy 集理论与 Fuzzy 逻辑;Dynamic Fuzzy 集与 Dynamic Fuzzy 逻辑;
- (3) 分布智能、群体智能、网络智能;神经网络模型与新算法;
- (4) 人工生命;演化计算;计算智能;
- (5) 数据挖掘;知识发现;机器学习;知识工程;知识理论;
- (6) 信息 Granular 与 Granulation 计算;
- (7) 自然语言理解与机器翻译;
- (8) 智能控制、智能管理;
- (9) 软计算理论及其应用;
- (10) 可拓工程理论及其应用;
- (11) 其他.

二、征文要求

论文必须未公开发表过,一般不超过 6000 字;论文一律用 A4 打印稿,一式 3 份,用 Word 7.0 以上排版,欢迎电子投稿;大会录用论文将在《计算机科学》杂志上发表.

三、重要日期

截稿日期: 2002 年 6 月 10 日 录用日期: 2002 年 7 月 10 日

四、联系方式

投稿方式: 215006,江苏省苏州市苏州大学计算机工程系 李凡长 教授

电子投稿请送: lfzh@suda.edu.cn

联系电话: 0512-5113205(O)(石明芳,王根荣)

李凡长教授电话: 013962116494;0512-5112737(办);0512-7489202(宅)