

# 基于脊线跟踪的指纹核心点检测<sup>①</sup>

王水鱼<sup>1</sup>, 刘武<sup>1</sup>, 王卓然<sup>2</sup>

<sup>1</sup>(西安理工大学 自动化与信息工程学院, 西安 710048)

<sup>2</sup>(西安理工大学 计算机科学与工程学院, 西安 710048)

**摘要:** 准确、可靠地检测指纹核心点对于指纹的分类和匹配有重要的意义。针对指纹图像核心点提取中准确判断和精确定位的难题,介绍了一种比较好的核心点检测算法。根据核心点是纹线曲率最大的点的定义,利用脊线跟踪的方法求出核心点。实验结果证明该方法能够从指纹图像中较精确、可靠地提取出核心点,并具有较好的鲁棒性。

**关键词:** 核心点; 脊线跟踪; 纹线曲率

## Core Point Detection of Fingerprint Based on Tracing Ridges

WANG Shui-Yu<sup>1</sup>, LIU Wu<sup>1</sup>, WANG Zhuo-Ran<sup>2</sup>

<sup>1</sup>(School of Automation and Information Engineering, Xi'an University of Technology, Xi'an 710048, China)

<sup>2</sup>(School of Computer Science, Xi'an University of Technology, Xi'an 710048, China)

**Abstract:** Accurate and reliable detection of core point for fingerprint classification and matching are important. This paper introduces a good method for core point detection in fingerprint to improve accuracy of the position and reliability of the core point. According to the definition of the core point is that the greatest point of curvature ridge, this paper use the ridge line tracing methods derive the core point. The experimental results show that the method can extract the core point from the fingerprint image more accurately and reliably, and has good robustness.

**Keywords:** core point; ridge line tracing; ridge curvature

## 1 引言

鉴于其惟一性、不变性和便携性等优点,指纹已成为当前应用最广泛的生物识别技术。自动指纹识别系统主要由指纹采集、图像预处理、特征提取、分类和匹配等部分组成。指纹核心点的定位是自动指纹识别技术中重要而基本的问题之一。核心点被定义为纹线上曲率最大的点,位于指纹纹路的渐进中心。在指纹特征匹配过程中,是否能精确地定位出指纹中心点,决定了特征匹配的准确程度。

指纹核心点的检测面临两个问题:一个是核心点的定位精度问题,另一个是可靠性问题。对于传统的指纹核心点检测算法存在漏检、偏检以及耗时过长的缺点,本文提出了一种复合的指纹核心点定位算法,首先对指纹图像进行预处理,然后对细化后的脊线进行跟踪并求出每条脊线上曲率最大的点,并按曲率大小

选出五至十个点,最后用这些点精确定位指纹核心点,实验结果表明该方法是有效、可靠的。

## 2 算法描述

### 2.1 指纹图像预处理

指纹图像的预处理的目的在于突出指纹图像中的脊线,消除图像中无用的干扰信息,以便于特征提取和识别。指纹预处理一般由方向图计算、滤波、二值化、细化等环节组成。

#### 2.1.1 方向图的计算

方向图描述了指纹图像中每一像素点所在脊线或谷线在该点的切线方向,分为两种:一种是点方向图,表示源指纹图像中每一点脊线的方向;另一种是块方向图,表示源指纹图像中每一块脊线的大致方向。

将指纹纹线的方向在 $0^\circ \sim 180^\circ$ 之内划分为*i*个基

① 收稿时间:2010-09-15;收到修改稿时间:2010-11-16

本方向,理论上*i*值越大越好,经试验及比较结果,*i*取8就可以了,*i*再取大不仅作用效果不大,相反还会消耗更多计算时间,因此我们把指纹脊线的走向分为八个方向,如表1所示1~8分别表示以P为中心的八个方向,表中标数字的地方为各方向对应的位置,空格表示在计算时无需考虑该像素的灰度值。为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*Gmean[i]*(*i*=0,□,7代表8个方向),*Gmean[i]*的求法如下:求取P的9×9窗口内*i*方向上与图中*i*位置对应的像素值的和,因为每个方向上取了四个采样点,所以除以4得到平均值。再将这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]*=|*Gmean[i]*-*Gmean[i+4]*|(*j*=0,...,3代表脊线方向),取差值的绝对值最大的两个方向*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*,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1。

表1 一个像素处的8个指纹脊线方向

|   |  |   |   |   |   |   |  |   |
|---|--|---|---|---|---|---|--|---|
| 2 |  | 3 |   | 4 |   | 5 |  | 6 |
|   |  |   |   |   |   |   |  |   |
| 1 |  | 2 | 3 | 4 | 5 | 6 |  | 7 |
|   |  | 1 |   |   |   | 7 |  |   |
| 0 |  | 0 |   | P |   | 0 |  | 0 |
|   |  | 7 |   |   |   | 1 |  |   |
| 7 |  | 6 | 5 | 4 | 3 | 2 |  | 1 |
|   |  |   |   |   |   |   |  |   |
| 6 |  | 5 |   | 4 |   | 3 |  | 2 |

为了减少噪声的影响,保证方向场估计的准确性,本文将图像分成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。

2.1.2 指纹图像滤波

指纹图像的滤波目的是去除指纹图中的叉连、断点及模糊之处。本文利用前面获取的方向图,设计了一组方向自适应滤波器,用以连接断点,去除叉点。方向自适应滤波器是由多个方向的滤波模板组成的,先设计出水平方向的滤波模板,其他方向的滤波模板可以通过旋转得到。

设水平方向滤波器大小为n×n,n通常由指纹图像中脊线和谷线的宽度决定,而脊线的宽度一般是3~10个像素,这里取n=7。模板的系数分布如表2所示。

表2中出现的变量是滤波器模板的系数,系数按*u > x > y > 0, u + 2x + 2y - 2z = 0*的关系确定<sup>2</sup>,得到水平方向的滤波器后,其他方向(2~8)的滤波器可通过旋转得到。

根据式(1)和每个像素的方向值,选用相应的滤波器模板进行卷积运算。

$$f(x,y) = \sum_{i=-3}^{i=3} \sum_{j=-3}^{j=3} G(x+i,y+j)g^\theta \tag{1}$$

其中,*g<sup>θ</sup>(i,j)*为相应的滤波器模板的系数,*G(x,y)*是原始图像的灰度值,*f(x,y)*为滤波后的图像。经过上述滤波处理后,一些断点和叉点想象被消除了,纹线得到了增强。

表2 水平方向滤波器模板

|      |       |    |    |    |       |      |
|------|-------|----|----|----|-------|------|
| -z/3 | -2z/3 | -z | -z | -z | -2z/3 | -z/3 |
| y/3  | 2y/3  | y  | y  | y  | 2y/3  | y/3  |
| x/3  | 2x/3  | x  | x  | x  | 2x/3  | x/3  |
| u/3  | 2u/3  | u  | u  | u  | 2u/3  | u/3  |
| x/3  | 2x/3  | x  | x  | x  | 2x/3  | x/3  |
| y/3  | 2y/3  | y  | y  | y  | 2y/3  | y/3  |
| -z/3 | -2z/3 | -z | -z | -z | -2z/3 | -z/3 |

2.1.3 图像二值化

图像二值化是将灰度图像变成0、1取值的二值图像过程。它设定一个阈值,当点的灰度值大于阈值时将该点设为1,灰度值小于阈值时将点设为0,这样就将整幅图像转化为由0和1组成的二值图像。本文采用自适应阈值法,将指纹图像分成若干个16×16的子区,先计算各子区的灰度均值,以此均值作为相应子区的阈值。再判断该子区中各个像素点,若像素值大于阈值,则置该点像素值为1,否则为0。另外,为防止背景块根据这种方法求出的阈值过低而引入噪声,加入一个判断条件,即如果某块的阈值低于某一固定值,则该块二值化的值应为0,这样就可以保证背景块能够从前景块中分离出来。

2.1.4 图像细化

指纹图像二值化以后,纹线仍具有一定的宽度,而指纹识别只对纹线的走向感兴趣,不关心它的粗细。为了进一步压缩数据,提高识别的准确性,需要对指纹图

像进行细化处理。指纹图像的细化是指删除指纹纹线的边缘像素，使之只有一个像素宽度，细化时应保证纹线的连接性、方向性和特征点不变，还应保持纹线的中心基本不变。本文采用模板细化法。根据指纹的纹理特征分别设置8个消除模板和6个保留模板<sup>3</sup>。细化从图像的第二条扫描线的第二个元素开始进行。用模板按行扫描二值图像，分别与8个消除模板比较，如果和每一个消除模板都不匹配时，该像素保留；否则，将抽取的像素再和六个保留模板进行比较，如果与其中的任一个模板匹配的话，则该像素保留，否则将其删除。重复这一过程，直到没有一个像素的值被改变为止。模板细化法能够很好的对指纹图像进行细化。

表3 消除模板

|   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | x | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | x | 0 |
| 0 | 1 | 1 | x | 1 | x | x | 1 | x | 1 | 1 | 0 |
| 0 | x | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | x | 0 |

|   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | x | 0 | 0 | 0 | x | x | 1 | x | x | 1 | x |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | x | 0 | x | 1 | x | 0 | 0 | x | x | 0 | 0 |

表4 保留模板

|   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|
| x | 1 | x | 0 | x | x | 0 | 0 | x | x | 1 | x |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| x | 1 | x | 0 | x | x | 1 | x | x | x | 0 | 0 |
| x | x | x | x | x | x | x | x | x | x | x | x |

|   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|
| x | 0 | x | 0 | x | 0 | x | x | x | 0 | x | x |
| 1 | 1 | 1 | x | x | 1 | x | x | x | 1 | x | x |
| x | 1 | x | x | 1 | 1 | 0 | x | 0 | 1 | 1 | x |
| 0 | 0 | 0 | x | x | 0 | 0 | x | 0 | 0 | x | x |

## 2.2 指纹脊线跟踪

细化的指纹二值图像，脊线为单像素宽，因此根据以每个值为1的像素点为中心的3×3窗口的模式特征来找出所有的正常脊线点、分叉点和端点。如图1所示，正常脊线点的3×3窗口里有3个像素值为1的点，分叉点的3×3窗口里有4个像素值为1的点，而端点的3×3窗口里则只有两个像素值为1的点。搜索整个指纹图像，把像素值为1且其3×3窗口中像素值为1的点的个数为3的点标记为正常脊线点，把像素

值为1且其3×3窗口中像素值为1的点的个数为4的点标记为分叉点，把像素值为1且其3×3窗口中像素值为1的点的个数为2的点标记为端点。搜索整个指纹图像，如找到一个像素值为1的点，且其标记正常脊线点，则从该点开始按该点的方向搜索其领域内的下一个正常脊线点，依次搜索直到碰到标记为分叉点或端点的点为止，然后再从开始点反方向搜索该脊线的其他脊线点，将找到的所有该脊线的像素点按顺序存储在一个链表中，这样就找到了一条脊线<sup>4</sup>。每找到一条脊线，就将刚找到的这条脊线上所有像素点的值修改为0，重新开始搜索下一条脊线，直到搜索完指纹图像中所有的脊线。

表5 正常脊线和特征点

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

(a) 正常脊线      (b) 分叉点      (c) 端点

## 2.3 核心点定位

### 2.3.1 脊线的曲率计算

对于平面曲线，曲率定义为曲线上某个点的切线方向角对弧长的转动率，通过微分来表示。但通过脊线跟踪获得的指纹脊线并不是平面曲线，虽然拥有与平面曲线相似的性质，但曲率的定义及计算方法并不完全一致。在这里，重新为曲率做如下定义<sup>5</sup>：

定义在点 $p_i$ 的左K斜率和的右K斜率分别为 $p_i$ 到 $p_{i-k}$ 方向和 $p_i$ 到 $p_{i+k}$ 方向的斜率。其中K为大于或等于1的整数。 $p_i$ 的左右K斜率的差，则定义为S曲线在点 $p_i$ 处的曲率。

$$S_k = \left| \frac{Y_{i-k} - Y_i}{X_{i-k} - X_i} - \frac{Y_{i+k} - Y_i}{X_{i+k} - X_i} \right| \quad (3)$$

$p_{i-k}$ 是从 $p_i$ 点出发沿着脊线向前边数到的和 $p_i$ 相隔K个像素的点，也即左斜率就是 $p_i$ 到 $p_{i-k}$ 的方向斜率， $p_{i+k}$ 、右斜率的定义与之类似。

从公式中可以看出，曲线上一点的曲率值并不是唯一的，而是随着用于计算的点位置不同而不同。在实际计算中，发现K的值大于等于3时，所得到的曲率值就与该点曲率的正确值基本近似。

### 2.3.2 核心点精确定位

在计算完每个点的曲率之后，为了排除掉端点和

分叉点带来的干扰,可以排除掉端点和分叉点,从正常脊线中选择出五到十个曲率最大的点作为候选点进行精确定位。

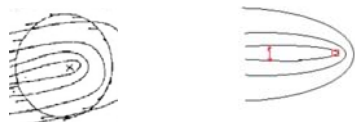


图1 核心点

从图中可以看出核心点的曲率是最大的,如果以核心点为中心以平行于脊线的直线为轴把其所在的脊线分成两半,则它的两部分脊线的距离是最小的。因此在得到候选点之后,可以以这些点为中心向两侧等间距搜索4~7对点,每对间隔3~5个像素,分别求得这几对点之间的距离,然后求平均值,则平均值最小的那条脊线认为是核心点所在的脊线,而那条脊线上曲率最大的点也就是核心点。

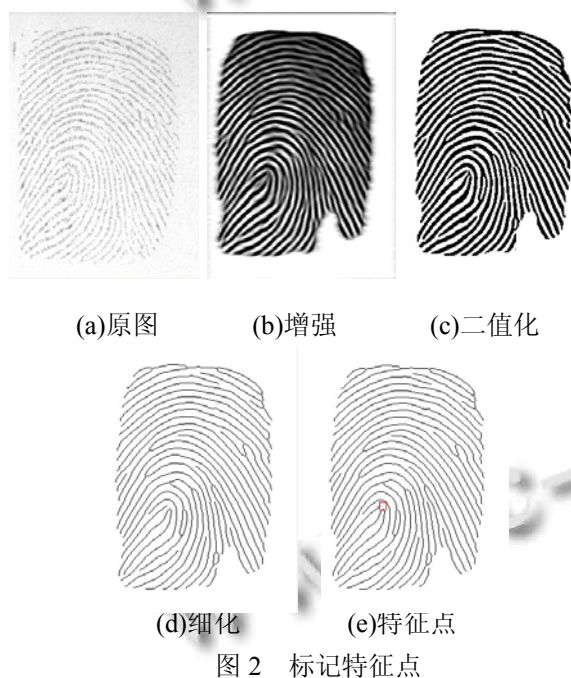


图2 标记特征点

### 3 实验结果及结论

衡量指纹核心点点检测算法性能的好坏的标准主

要是看这个算法能否准确地找到指纹图像中的核心点,不遗漏中心点,也没有出现伪中心点,同时对中心点的位置定位准确。

表6 本算法测试结果与算法结果 Poincare 对比

| 方案       | 错误数 | 正确率  |
|----------|-----|------|
| 本文方案     | 9   | 91.2 |
| Poincare | 13  | 87.4 |

通过大量的实验,证明了本文算法对不同类型的指纹图像适应性较好,且能较精确地定位指纹中心点,但对于一些低质指纹图像,算法还是很难准确地定位其核心点,今后将更进一步对方向场进行修正研究,以使算法更全面,鲁棒性更强。

### 参考文献

- 姚辰松.自动指纹识别技术研究[硕士学位论文].合肥:合肥工业大学,2008.
- 陈昌,徐晓明.指纹预处理中的图像增强.中国信息科技,2005,24.
- 戴春霞.基于改进指纹图像细化算法的识别系统[硕士学位论文].湖南大学,2004.
- 姜红超,张高伟.基于脊线跟踪的指纹细节特征提取.计算机工程,2007,33(8):185-186.
- 修照军.基于脊线矢量化指纹识别算法研究[硕士学位论文],2008.
- Karu K, Jain A. Fingerprint classification. Pattern Recognition, 1996,29(3):389.
- Nilsson K, Bigun J. Localization of corresponding points in fingerprints by complex filtering. Pattern Recognition Letters, 2003,24(13):2135-2144.
- 傅德生,王水平,李仕强.指纹中心自动定位诸方法研究及应用.计算机应用于软件,2004,21(3):10-13.
- 杨海军,梁德群.基于方向场特征的指纹图像奇异点检测.自动化学报,2001,13(2):272-275.
- 韩智,刘昌平.一种鲁棒的指纹奇异点检测方法.计算机工程,2006,32(20):30-32.