

一种鲁棒的指纹奇异点检测方法

韩 智^{1,2}, 刘昌平¹

(1. 中国科学院自动化研究所, 北京 100080; 2. 中国科学院研究生院, 北京 100039)

摘要: 提出一种两阶段的奇异点检测方法。将指纹图像分块, 求出各块的方向构成块方向图, 并在块方向图的基础上利用邻域方向的分布分析结合改进的 Poincaré Index 方法来确定奇异点所在的候选区域, 对候选区域中的像素再通过计算局部方向变化率来确定奇异点的精确位置。将此方法用于对 FVC2004 DB1_A 指纹数据库的图像, 实验结果表明这种方法对指纹图像中的噪声有很好的鲁棒性, 并且计算简单快速, 易于实现。

关键词: 指纹识别; 奇异点; 方向场; 局部方向变化率

A Robust Method of Singular Point Detection from Fingerprint Image

HAN Zhi^{1,2}, LIU Changping¹

(1. Institute of Automation, Chinese Academy of Sciences, Beijing 100080; 2. Graduate School of Chinese Academy of Sciences, Beijing 100039)

【Abstract】 A two-stage robust algorithm of singular points detection based on orientation features of fingerprint images is proposed. Candidate small blocks which may contain singular points are located with the integrated method of improved Poincaré index calculation with the orientation distribution analysis of the block orientation images; Local orientation variance in the candidate blocks is computed and used to locate the position of singular points accurately. The algorithm is evaluated with images from FVC2004 DB1_A database and the experimental results prove the method accurate and efficient.

【Key words】 Fingerprint identification; Singular point; Orientation image; Local orientation variance

1 概述

指纹因其具有唯一性和稳定性, 一直是身份鉴定的可靠手段。近年来, 有关指纹自动识别的研究已成为模式识别、图像理解及计算机视觉等领域中广为关注的热点。

指纹的奇异点包括两种类型: 一类是中心点(core point), 另一类是三角点(delta point), 如图 1(a)所示。指纹的奇异点的数目、类型和位置等信息经常被用来解决指纹的分类和指纹匹配中的指纹对齐问题。因此, 准确、可靠地检测奇异点的位置, 对于自动指纹识别系统具有重要的意义。



图 1 指纹图像的奇异点和方向图
○ 标记中心点的位置, △ 标记三角点的位置

图 1 指纹图像的奇异点和方向图

指纹奇异点的检测要解决好两个问题: 一个是奇异点的定位精度问题, 另一个是可靠性问题。目前奇异点检测的主流方法是最早由 Kawagoe and Tojo^[1]提出的通过分析每个点的 Poincaré 索引值来确定指纹的奇异点^[1,2]。Poincaré 索引值是指以某个点为中心的一条封闭曲线上各点的方向角度变化量之和, 对指纹上不同类型的点的 Poincaré 索引值不同, 非奇异点的为 0, 三角点为 $-\pi$, 漩涡型中心点为 2π , 环型中心点为 π 。这种方法对噪音比较敏感, 在质量较差的图像中容易产生伪奇异点。Asker M. Bazen 等人提出一种通过检测纹

线方向变化梯度的方法来检测奇异点^[3]。V.S. Srinivasan 等提出通过求每个点周围的 4 个象限区域的方向图中的主导方向并结合一些规则来检测奇异点的存在和类别^[4]。Capelli 等提出一种用表示点周围方向场的局部偏差的不规则算子来提取奇异点的方法^[5]。张伟伟等提出了通过角点检测奇异点的方法^[6]。

2 算法描述

2.1 方向场的计算

本文中奇异点检测的算法是基于指纹图像的方向场进行计算的。指纹的方向场是指图像中脊线结构的局部方向, 它描述了指纹的基本形状。本文采用了文献[7]中提出的基于梯度的求方向场的方法, 即通过计算图像的某一点的邻域的平均灰度梯度来计算这一点的方向场。

用 f 表示原始指纹图像, $f(i,j)$ 为图像中像素点 (i,j) 的灰度值。本文中采用如下的步骤计算指纹图像中像素点 (i,j) 处的方向:

(1) 选取以 (i,j) 为中心的大小为 $w \times w$ 的方形邻域, 按下公式计算点 (i,j) 的方向 $\theta(i,j)$:

$$\theta(i,j) = \frac{1}{2} \tan^{-1} \left(\frac{\sum_{u=i-w/2}^{i+w/2} \sum_{v=j-w/2}^{j+w/2} 2G_x(u,v)G_y(u,v)}{\sum_{u=i-w/2}^{i+w/2} \sum_{v=j-w/2}^{j+w/2} (G_x^2(u,v) - G_y^2(u,v))} \right)$$

其中 $G(u,v) = \begin{bmatrix} G_x(u,v) \\ G_y(u,v) \end{bmatrix}$ 为点 (u,v) 的梯度向量, 通过 Sobel 算

作者简介: 韩 智(1972 -), 男, 博士生, 主研方向: 生物特征识别, 文字识别, 模式识别, 图像处理; 刘昌平, 研究员, 博导

收稿日期: 2006-01-10 **E-mail:** zhi.han@ia.ac.cn

子来进行计算。

(2)将 $\theta(i, j)$ 按下式归一化 12 个方向值, 记为 $o(i, j)$, 即

$$o(i, j) = (\theta(i, j) / \pi * 12) \bmod 12$$

$o(i, j)$ 即构成了指纹的方向图。图 1(b)给出了一个指纹图像的方向图的示例显示, 为了显示出不同的指纹方向的变化, 每个点显示的像素值为 $o'(i, j)$, 其中 $o'(i, j) = o(i, j) * 20$ 。

2.2 奇异点的粗定位

从图 1(b)中可以看到, 指纹的奇异点附近的方向变化非常剧烈, 表现为在奇异点的邻域内出现的指纹方向数目多, 并且 12 个方向的分布比较均匀; 而在非奇异点邻域内出现的方向数目相对单一, 并且 12 个方向的分布比较单一。

指纹奇异点的粗定位是指快速地找到奇异点所在的区域, 在不影响准确性的情况下有效地提高指纹奇异点检测算法的速度。本文选择以块方向图为基础进行奇异点区域的粗定位, 即把图像分为不重叠的若干个子块, 以每个子块图像中心像素的方向作为该子块的方向, 所有子块的方向即构成指纹的块方向图。本文采用传统的 Poincaré 指标方法结合邻域方向分布的分析来确定奇异点所在的区域。奇异点所在区域定位的具体方法步骤如下:

(1)将指纹图像分成互不重叠的大小为 $m \times m$ 的子块。设指纹图像的大小为 $M \times M$, 则指纹图像被分为 $N_B \times M_B$ 个子块, 其中 $N_B = M/m$, $M_B = M/m$ 。本文中实验所用的 500dpi 的指纹图像选取 $m=16$ 。用 $B_{(i,j)}$ 来表示以 (i,j) 为中心的子块图像。

(2)计算每个子块图像的方向。将每个子块的中心像素的方向作为整个子块图像的方向, 即 $o(B_{(i,j)}) = o(i, j)$, 其中 $o(B_{(i,j)})$ 表示子块图像 $B_{(i,j)}$ 的方向, $o(i, j)$ 为点 (i, j) 处的方向, 按 2.1 节中的方法进行计算。所有子块的方向即构成一个大小为 $N_B \times M_B$ 的块方向图, 记为 O_B 。

(3)统计块方向图 O_B 中每个像素的 8 邻域像素中出现的不同的方向值个数, 记为 $N_{Direction}$ 。如果 $N_{Direction} < M$, 则认为这个区域中不会出现奇异点。 M 为预先设定的阈值, 用来判断奇异点是否出现在这个子块区域中。本文中选取 $M=4$ 。记集合 $S = \{c | N_{Direction}(c) \geq M, c \in O_B\}$

(4)计算 S 中所有像素的 Poincaré 索引值。为克服 Poincaré 索引方法对噪声敏感的缺点, 选取两条封闭曲线分别计算 Poincaré 索引值。两条封闭曲线在 5×5 的方格中选取, 如图 2 所示。则按如下公式计算给定点 (i, j) 在封闭曲线 $D_0 D_1 \dots D_{15} D_0$ 上的 Poincaré 索引值 $PI_1(i, j)$ 。

$$PI_1(i, j) = \sum_{k=0}^{15} \Delta(k)$$

$$\text{其中 } \Delta(k) = \begin{cases} \delta(k) & |\delta(k)| < 6 \\ 12 + \delta(k) & \delta(k) < -6 \\ 12 - \delta(k) & \text{otherwise} \end{cases}$$

$\delta(k) = o(D_k) - o(D_{(k+1) \bmod 16})$, $o(D_k)$ 表示点 D_k 的方向。

D ₆	D ₅	D ₄	D ₃	D ₂
D ₇	d ₃	d ₂	d ₁	D ₁
D ₈	d ₄	(i,j)	d ₀	D ₀
D ₉	d ₅	d ₆	d ₇	D ₁₅
D ₁₀	D ₁₁	D ₁₂	D ₁₃	D ₁₄

图 2 计算 Poincaré 索引所选取的封闭曲线示意

类似地, 可以计算给定点 (i, j) 在封闭曲线 $d_0 d_1 \dots d_7 d_0$ 上的 Poincaré 索引值 $PI_2(i, j)$ 。

本文按如下条件选取 O_B 中候选奇异点集合, 并记为 S_{SP} :

$$S_{Core} = \{(i, j) | PI_1(i, j) \geq 10 \& PI_2(i, j) \geq 10, (i, j) \in O_B\}$$

$$S_{Delta} = \{(i, j) | PI_1(i, j) \leq -10 \& PI_2(i, j) \leq -10, (i, j) \in O_B\}$$

$$S_{sp} = S_{Core} \cup S_{Delta}$$

其中 S_{Core} 与 S_{Delta} 分别为 O_B 中候选中心点和候选三角点的集合。 O_B 所有属于 S_{SP} 中的点在原始指纹图像中所对应的子块图像即为奇异点候选子块图像, 则指纹图像中奇异点的候选区域 S_{CBR} 可以用这些子块图像构成的集合来表示, 即 $S_{CBR} = \{B_{(i_1, j_1)}, B_{(i_2, j_2)}, \dots, B_{(i_{N_s}, j_{N_s})}\}$, 其中 N_s 为候选子块图像的个数。

2.3 奇异点的精确定位

在确定了奇异点的候选区域后, 需要进一步从候选区域中来确定奇异点的精确位置。从图 1(b)的指纹方向图中可以发现指纹的奇异点处是其局部所在区域中方向变化最剧烈的地方, 因此可以通过计算候选区域的局部方向变化率的最大值来确定指纹图像奇异点的位置。奇异点的精确定位的具体步骤如下:

(1)扩展候选子块图像区域: 将 S_{CBR} 中的每个候选子块图像的区域进行扩展, 将 $B_{(i_k, j_k)}$ 扩展为以 (i_k, j_k) 为中心的大小为 $\frac{3}{2}w \times \frac{3}{2}w$ 的子块区域, 记为 $B'_{(i_k, j_k)}$, 将扩展后的候选区域仍记为 S_{CBR} , 即 $S_{CBR} = \{B'_{(i_1, j_1)}, B'_{(i_2, j_2)}, \dots, B'_{(i_{N_s}, j_{N_s})}\}$ 。由于在粗定位过程中使用块方向图计算, 当奇异点处于子块图像的边缘附近时, 使用块方向图计算 Poincaré 指标可能会出现偏差, 因此需要对子块图像区域进行扩展来更准确地确定奇异点的位置, 避免遗漏奇异点。

(2)计算候选子块图像的方向图: 对每个候选子块图像, 按 2.1 节中的方法求出区域中每个点的方向, 构成一个子块区域的方向图。

(3)计算局部方向变化图: 对子块方向图中的每个点, 按如下公式求此像素点的局部方向变化率 $V(P)$ 。

$$V(P) = \sum_{k=0}^7 |o(P) - o(NP(k))|$$

$O(P)$ 表示点 P 的方向, $NP(k)$ 表示给定点的第 k 个 8 邻域像素点。所有的局部方向变化率组成了一个候选子块区域的局部方向变化图。

(4)确定候选奇异点: 在每个扩展后的候选子块图像区域中, 具有最大局部方向变化率的像素被认为是一个候选奇异点。

(5)确定奇异点的类型: 对某个候选奇异点 P , 设其所在子块图像为 B_p , 如果 $B_p \in S_{Core}$, 则 P 为中心点; 如果 $B_p \in S_{Delta}$, 则 P 属于三角点。

(6)检查所有候选奇异点, 如果两个坐标分别为 (i_1, j_1) 和 (i_2, j_2) 的候选奇异点 P_1 和 P_2 满足 $|i_1 - i_2| < W$ 且 $|j_1 - j_2| < W$ 并设 $V(P_1) > V(P_2)$, 则删除候选奇异点 (i_2, j_2) 。这一步主要是考虑奇异点邻近的点的方向变化也比较剧烈, 并且奇异点的邻近的点可能与奇异点处于不同的子块从而在其它的子块中被选做候选奇异点。通过这一步可以去掉这类假的奇异点。

(7)检查所有的候选奇异点, 如果类型为中心点的候选奇异点数目大于 2, 则选择局部方向变化率最大的 2 个作为图像的中心点; 同样, 若类型为三角点的候选奇异点数目大于 2, 则选择方向变化率最大的 2 个作为图像的三角点。

3 实验结果和结论

衡量指纹奇异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奇异点,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。为测试本文提出的奇异点检测算法的性能,选择FVC2004^[8]的DB1数据库中的图像对本文中指纹奇异点检测的算法进行了测试。DB1_A指纹数据库中包括了100个人的800幅不同质量、不同类别的指纹图像。在对本文提出的算法的测试中,对各种不同的错误如遗漏奇异点、伪奇异点和奇异点位置不准确的情况进行了统计,表1给出了测试结果。图3给出了对不同类型的指纹图像用本文的方法提取奇异点的结果。

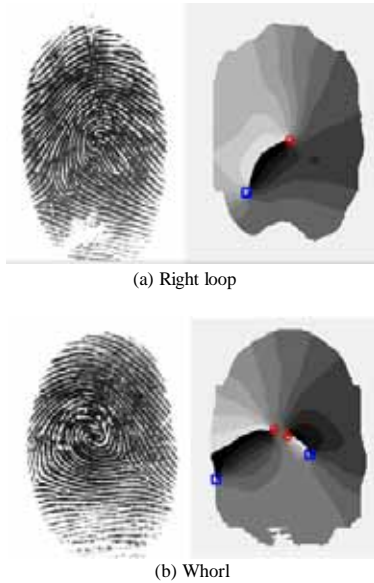


图3 部分不同类型的指纹图像的奇异点提取结果

本文提出了一种指纹奇异点提取方法,首先在块方向图的基础上利用指纹方向的分布和改进的Poincaré方法来确定奇异点所在的候选区域,再通过求候选区域内点的局部方向变化率来精确确定指纹奇异点的位置。利用指纹方向分布与Poincaré方法相结合进行候选区域的定位可以有效地避免单纯使用Poincaré方法对噪音或低质量指纹图像敏感的弱点,同时局部方向的变化可以有助于准确地定位奇异点的位置。

从图3和表1的实验结果表明,这种方法能够比较准确地定位指纹图像中的奇异点,对噪声有很好的鲁棒性,并且计算简单快速,易于实现。

表1 本文中的算法在FVC2004的DB1数据库上的测试结果

奇异点数目	1 279
中心点数目	921
三角点数目	358
遗漏中心点数目	37
遗漏三角点数目	4
伪中心点数目	25
伪三角点数目	3
奇异点位置不准确数目	23
总错误数	92
正确率	92.81%

通过对实验结果中错误情况的分析发现,很多错误是由于错误的方向场计算造成的,今后在对低质量指纹图像的方向场计算上还需进一步的改进。

参考文献

- 1 Kawagoe M, Tojo A. Fingerprint Pattern Classification[J]. Pattern Recognition, 1984, 17(3): 295-303.
- 2 Karu K, Jain A K. Fingerprint Classification[J]. Pattern Recognition, 1996, 29(3): 389-404.
- 3 Bazen A M, Gerez S H. Systematic Methods for the Computation of Directional Fields and Singular Points of Fingerprints[J]. IEEE Transactions on PAMI, 2002, 24(7).
- 4 Srinivasan V S, Murthy N N. Detection of Singular Points in Fingerprint Images[J]. Pattern Recognition, 1992, 25(2): 139-153.
- 5 Cappelli R, Lumini A, Maio D, et al. Fingerprint Classification by Directional Image Partitioning[J]. IEEE Transactions on PAMI, 1999, 21(5): 402-421.
- 6 张伟伟, 王 森, 王阳生. 基于角点检测的指纹图像奇异性检测 [C]. 第6届国际青年计算机工作者会议暨第2届生物特征识别研讨会会议论文集, 杭州, 2001, S4: 51-56.
- 7 Kass M, Witkin A. Analyzing Oriented Patterns[J]. Computer Vision, Graphics, and Image Processing, 1987, 37(3): 362-385.
- 8 FVC2004: Third Fingerprint Verification Competition[Z]. <http://bias.csr.unibo.it/fvc.2004>.

(上接第27页)

是目标颜色特征,但同样可以使用纹理等特征,算法思想、步骤一样,都是用改进直方图映射算法构建灰度图,在灰度图中使用均值移位算法定位目标。文中算法可以实时跟踪目标,目标在一个矩形框或者椭圆内,但不能得到不规则目标轮廓。主动轮廓算法可以准确得到目标不规则轮廓,缺点是初始化曲线距离目标太远时效果不好。而我们的算法已经能跟踪目标区域,所以可以在矩形框中使用主动轮廓算法将得到目标的精确轮廓。

参考文献

- 1 Comaniciu D, Ramesh V, Meer P. Kernel-based Object Tracking[J]. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2003,

25(5): 564-577.

- 2 Cheng Y, Shift M. Mode Seeking and Clustering[J]. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1995, 17(8): 790-799.
- 3 Bradski G R. Computer Vision Face Tracking for Use in a Perceptual User Interface[J]. Intel Technology Journal, 1998, 5(1): 58-76.
- 4 Lee J H, Lee W H, Jeong D S. Object-tracking Method Using Back-projection of Multiple Color Histogram Models[J]. Circuits and Systems, 2003, 2(2): 668-671.
- 5 Fukunaga K, Hostetler L D. The Estimation of the Gradient of a Density of Function, with Applications in Pattern Recognition[J]. IEEE Trans. on Information Theory, 1975, 21(1): 32-40.