

指纹图像预处理和特征提取

刘灵丽^{1,2}, 李丽娟¹

(1. 湖南大学计算机与通信学院, 长沙 410082; 2. 湘南学院计算机系, 郴州 423000)

摘要: 根据指纹的固有规律, 提出了一套较完整的指纹图像预处理和特征提取算法。改进了指纹脊线频率求取算法和二值图像去孔噪声算法, 并提出了一种新的滤除指纹伪特征点的方法。通过对上百幅不同质量的指纹图像进行测试, 获得了较好的效果。

关键词: 指纹; 预处理; 特征提取

Preprocessing and Minutiae Extraction of Fingerprint Image

LIU Lingli^{1,2}, LI Lijuan¹

(1. College of Computer and Communication, Hunan University, Changsha 410082;

2. Department of Computer, Xiangnan University, Chenzhou 423000)

【Abstract】 A set algorithm for preprocessing and minutiae extraction of fingerprint image is proposed in this paper. It is based on the analysis of the inherent feature of fingerprint. The method to get ridge frequency and the method to eliminate holes of binary image are proposed. Besides, a new approach to eliminate false minutiae of thinned fingerprint image is presented. These proposed algorithms have been widely tested and checked for hundreds of fingerprint images and excellent results are obtained.

【Key words】 Fingerprint; Preprocessing; Minutiae extraction

自动指纹识别技术一般是基于指纹的细节特征提取和匹配的。指纹的细节特征主要指脊线端点和分叉点。端点和分叉点的提取精度和匹配程度决定指纹识别系统的性能好坏。由于多种因素的影响, 在提取指纹特征前必须对输入的指纹图进行预处理, 预处理的目的是: 去除原图像中的噪声, 把它变成一幅清晰的二值点线细化图, 以便于提取正确的指纹特征。然而, 由于图像本身的质量问题和预处理过程中也可能带来某些噪声, 从细化后的指纹图像中直接提取的细节特征还存在一些伪特征。本文提出了一套较完善的指纹预处理算法, 包括规格化、图像分割、脊线方向计算、脊线频率计算、图像增强、二值化、二值滤波和细化等步骤; 改进了指纹脊线频率求取算法和二值图像去除孔噪声算法, 并给出了一种较实用的滤除伪特征的方法。

1 预处理算法

1.1 规格化和图像分割

规格化是把不同原图像的平均灰度和对比度调整到一个固定的级别, 为后续处理提供一个较为统一的图像规格。本实验使用的指纹输入设备是 Veridicom 公司推出的第 3 代半导体指纹传感器 FPS200。获取的图像是 256 级灰度图像, 大小是 256 × 300 像素, 分辨率为 500dpi。实验中期望灰度均值和方差分别取 140、600。图像分割是把指纹前景区与背景区分开。指纹图像的前景区是由脊和谷交替组成的, 其灰度统计特性中局部灰度方差比较大, 而指纹图像的背景区, 这一值是很小的。基于这一特性, 可以利用图像的局部方差对指纹图像进行分割。首先把图像划分为互不重叠的 W × W (本实验取 8 × 8) 的子块, 计算每块的均值和方差, 设定阈值 T (本实验取 30), 方差大于 T 的子块定义为前景区, 否则为背景区。

1.2 脊线方向计算

指纹图像的脊线方向表明指纹的内在特性, 在指纹图像分析中起着重要作用, 目前有很多求脊线方向的算法, 本文采用如下算法^[1]:

- (1) 将图像分成互不重叠的 W × W (本实验取 15 × 15) 的子块;
- (2) 计算块中每个像素的梯度 G_x 和 G_y;
- (3) 计算每块图像的脊线方向:

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

1.3 脊线距离和频率计算

在指纹图像的前景区, 沿着垂直于脊线的方向, 图像灰度分布近似为正弦波, 该正弦波的频率可视为指纹局部区域的脊线频率。它和脊线方向一样, 也是指纹图像的一个重要属性。利用脊线频率和方向, 可以有效地对指纹图像进行滤波增强。脊线频率的计算方法如下^[1]:

(1) 对指纹图像中的任意一个子块(i, j)(i, j为子块图像中心的横坐标和纵坐标), 建立一个以像素点(i, j)为中心的方向窗口 N × S (本文取 33 × 17), 其窗口的边框与指纹脊线对应, 且 N 与脊线方向垂直, S 与脊线方向平行, 如图 1 所示。

基金项目: 国家自然科学基金资助项目(60373062); 教育部科研基金资助重点项目(03092); 湖南省杰出中青年基金资助项目(02JJYB012)

作者简介: 刘灵丽(1977—), 女, 硕士生, 主研方向: 图像处理, 模式识别; 李丽娟, 教授

收稿日期: 2005-09-14 **E-mail:** hudajty2003@yahoo.com.cn

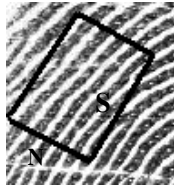


图1 方向窗口

(2)对于每一子块 (i,j) ,沿 S 方向计算 S 个像素的灰度均值 $X(k)$,沿 N 方向上共统计 N 个 $X(k)$,计算方法如下:

$$X(k) = \frac{1}{S} \sum_{d=0}^{S-1} G(u,v), k = 0, 1, 2, \dots, N-1$$

$$u = i + (d - \frac{S}{2}) \cos(\theta(i,j)) + (k - \frac{N}{2}) \sin(\theta(i,j))$$

$$v = j + (d - \frac{S}{2}) \sin(\theta(i,j)) - (k - \frac{N}{2}) \cos(\theta(i,j))$$

式中 $\theta(i,j)$ 为图像子块方向,即脊线方向。

(3)计算脊线距离。对于有效的指纹图,第(2)步中求的 $X(k)$ 应是一离散正弦波,设 $T[i,j]$ 为 $X(k)$ 中两个连续峰值之间的平均像素个数,则 $T[i,j]$ 为局部脊线距离。由于指纹图像质量的问题,上面求得的局部脊线距离可能不准,Hong等人^[1]将脊线距离变动范围规定为 $[3,25]$ 。本实验经统计发现:对于从某个具体的指纹采集仪采集到的一系列图像,它们有效的脊线距离范围远远小于 $[3,25]$,因此文献^[1]对局部脊线距离的平滑作用太大,影响了后续指纹图像增强的效果。针对此问题,本实验改进了局部脊线距离的求取算法,改进算法如下:

- 1)求指纹图像前景区中各像素点的局部脊线距离;
- 2)对局部脊线距离进行直方图统计,求出直方图中前7个峰值 T_1, \dots, T_7 ,令: $T_{\max} = \text{MAX}(T_1, \dots, T_7)$; $T_{\min} = \text{MIN}(T_1, \dots, T_7)$;
- 3)检查各个像素点的局部脊线距离 $T[i,j]$,
if $(T[i,j] > T_{\max})$ then $T[i,j] = T_{\max}$;
if $(T[i,j] < T_{\min})$ then $T[i,j] = T_{\min}$;
(本实验 T_{\max} , T_{\min} 分别取经验值11和5)。
- 4)局部脊线频率 $F(i,j) = 1/T[i,j]$, $T[i,j]$ 为第(3)步求的局部脊线距离。

1.4 图像增强、二值化和二值滤波

人们对指纹图像的增强进行了大量研究,本实验采用目前公认效果最好的基于Gabor滤波器的指纹图像增强方法^[1]。二值化是把灰度图像变成二值图像,使指纹的脊线变得更加清晰,本文采用方向图二值化方法^[2]。

经过上述步骤,得到的二值图像脊线中间可能存在一些小孔,又称气泡。如不去除这些气泡而直接细化,在气泡处容易产生伪分叉,为特征提取和识别带来干扰,因此有必要对二值图像再进行一次滤波。文献^[3]采用了种子填充算法去除孔噪声。该算法效果较好,但其种子点选择条件为: $G(i,j) > G(i+1,j)$ 且 $G(i,j) > G(i,j+1)$, $G(i,j)$ 为点 (i,j) 的像素值;该条件过于宽松,因此文献^[3]中的算法对大量伪种子点进行了种子填充,当发现不合理时才取消该次种子填充,速度较慢。

本实验对大量二值指纹图像进行了观察与分析,发现有一规律:任何气泡所在区域至少有一个8邻域黑点数大于4的白点,因此,可得出如下结论:去除二值指纹图像孔噪声,只需要对8邻域黑点数大于4的白点进行种子填充即可。新的去除孔噪声算法如下:

- (1)按从上到下,从左到右的顺序,对图像中的白点计算其8邻域的黑点数 S_N ;
- (2)对 S_N 进行判断,当 $S_N > 4$,认为该白点是一个气泡的种子点,将它压入堆栈;

(3)从堆栈弹出一个像素点,把它作为当前点,将其填充为脊线上的点,并统计填充区域总点数 T ,若 $T > T_0$ (T_0 为一合适阈值,本实验取30),取消该次填充,否则填充有效;

(4)从堆栈中除去该种子点;

(5)判断堆栈中是否还有种子点,如有转(3),否则转(1)。

2 特征提取和滤除伪特征算法

2.1 特征提取算法

细化图像上点 N 的交叉数定义为

$$C_N = \frac{1}{2} \sum_{k=1}^8 |R(k+1) - R(k)|, R(9) = R(1)$$

其中 $R(1), R(2), \dots, R(8)$ 是细化后图像 N 点沿顺时针方向排列在 X_1, X_2, \dots, X_8 处的灰度,如图2。如 $C_N = 1$,点 N 为脊线端点,如 $C_N = 3$,点 N 为脊线分叉点。

X_8	X_1	X_2
X_7	N	X_3
X_6	X_5	X_4

图2 8邻域图

2.2 伪特征点的特点分析

细化图像中存在的噪声包括毛刺、短脊、断脊、假桥、小孔和岛屿^[5],分别如图3(a)~图3(f)。上述噪声结构产生的伪特征点可归类为:

- (1)由毛刺产生的伪端点和伪分叉点,伪端点和伪分叉点之间有脊线相连且距离较短;
- (2)由短脊产生的伪端点,伪端点之间有脊线相连且距离较短;
- (3)由断脊产生的伪端点,伪端点之间没有脊线相连且距离较短;
- (4)由假桥、小孔和岛屿产生的两个或多个伪分叉点,这些伪分叉点之间彼此距离一般较短。

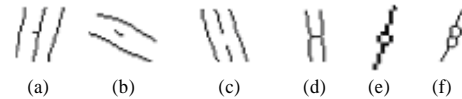


图3 细化图像中存在的噪声

2.3 滤除伪特征点算法

本实验通过分析伪特征点的特点,提出如下滤除伪特征点的算法:

- (1)检查所有特征点:若为边界点则滤除;否则,若为非边界端点则跟踪脊线,且如果在 D 步内找到了另一特征点(端点或分叉点),则认为这些特征点是由短脊或毛刺产生的,将它们滤除;
- (2)对剩下的特征点,以特征点为中心建立一个 $(2N+1) \times (2N+1)$ 的窗口:
 - 1)若特征点为端点,且在窗口内检测到另一个与该端点所在连线方向与脊线方向夹角小于 R 的端点,则认为这两个端点是由断脊产生的伪端点,将它们滤除;
 - 2)若特征点为分叉点,且在窗口内检测到其他的分叉点(一个或多个),则认为这些分叉点是由假桥、小孔或岛屿产生的伪分叉点,将它们滤除;
 - 3)将剩下的特征点作为真正的特征点集。

其中 D, N, R 是门限值,本实验分别取经验值14, 11, 16。

本算法先跟踪非边界端点,滤除毛刺和短脊产生的伪端点和伪分叉点,然后用窗口法,滤除由断脊、假桥、小孔和岛屿产生的伪特征点。与一般的脊线跟踪法相比,本算法能快而准确地滤除毛刺和短脊产生的伪端点和伪分叉点,而对剩下的由断脊、假桥、小孔和岛屿产生的伪特征点,考虑到这些伪特征点要么都是端点,要么都是分叉点,采用简单的窗口法,能很快地滤除这些伪特征点且一定能保留这些伪端点附近的真分叉点和伪分叉点附近的真端点;整个算法与常用算法^[5-7]相比计算量较小。

3 实验结果与分析

图 4(a)是一幅原始指纹图像,图 4(b)是规格化并图像分割后的图像,图 4(c)、图 4(d)分别是文献[1]和本实验图像增强后的结果,图 4(e)、图 4(f)分别是图 4(c)、图 4(d)矩形区的放大图,对比图 4(e)、图 4(f),可以看出,使用本实验改进后的脊线距离求取算法进行图像增强,得到的图像脊线更为流畅。

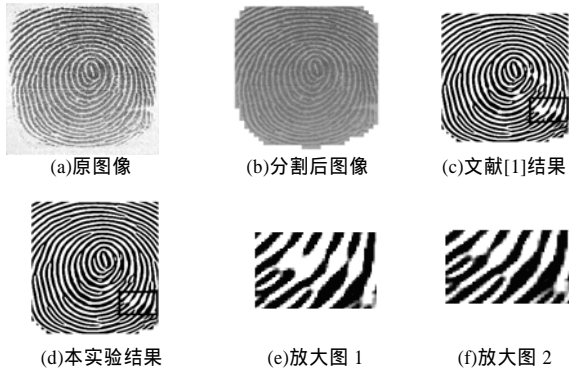


图 4 图像增强结果

图 5(a)是图 4(d)的二值化结果,图 5(b)是二值滤波结果,图 5(c)、图 5(d)分别是图 5(a)中含有较多小孔的矩形区滤波前和滤波后的放大图。对比图 5(c)、图 5(d),不难发现,改进算法能有效去除二值图像孔噪声。



图 5 二值滤波结果

图 5(e)、图 5(f)分别是图 5(a)中直线 1、2 左端点所在区

域的放大图,其中图 5(e)含有一气泡,图 5(f)不含气泡,图 5(g)、图 5(h)中的灰色小块分别代表图 5(e)、图 5(f)中符合文献[3]种子点条件的像素点,图 5(i)、图 5(j)中的灰色小块则分别代表图 5(e)、图 5(f)中符合本实验种子点条件的像素点。分别对比图 5(g)、图 5(i)和图 5(h)、图 5(j),可以发现,本实验的种子点个数明显少于文献[3]。编程计算图 5(a)中符合文献[3]和本实验的种子点个数分别是 972 和 66,因此本实验去孔噪声速度比文献[3]要快得多。

图 6(a)是用文献[4]方法对图 5(b)细化处理结果。图 6(b)是对图 6(a)直接提取细节特征的结果,图 6(c)是本实验滤除伪特征后的结果。从图中可以看出,该算法滤除伪特征点效果较好。



图 6 滤除伪特征结果

4 结束语

本文提出了一套完整的指纹图像预处理和特征提取算法,它利用了指纹的固有规律,改进了指纹脊线频率求取算法和二值图像去孔噪声算法,并提出了一种新的滤除指纹伪特征点的方法。从实验结果来看,整套算法比较令人满意。

参考文献

- Hong L, YiFei W, Jain A K. Fingerprint Image Enhancement: Algorithm and Performance Evaluation[J]. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1998, 20(8): 777-788.
- Candela G T, Gtother P J, Watson C I, et al. PCASYSA Pattern-level Classification Automation System for Fingerprints[R]. Technical Report, NISTIR 5467, 1995.
- 王 玮, 李见为, 张 腾. 指纹图像的预处理算法[J]. 计算机应用, 2004, 24(5): 72-75.
- 王家隆, 郭成安. 一种改进的图像模板细化算法[J]. 中国图像图形学报, 2004, 9(3): 297-301.
- Xiao Q, Raafat H. Fingerprint Image Postprocessing: A Combined Statistical and Structural Approach[J]. Pattern Recognition, 1991, 24(10): 985-992.
- Tico M, Kuosmanen P. An Algorithm for Fingerprint Image Postprocessing[C]. proc. of the Thirty-fourth Asilomar Conference, Signals, Systems and Computers, 2000: 1735-1739.
- Bian Z, Zhang D, Shu W. Knowledge Based Fingerprint Postprocessing[J]. International Journal of Pattern Recognition and Artificial Intelligence, 2002, 16(1): 53-67.

(上接第 189 页)

- Angeline P J, Pollack J B. Competitive Environments Evolve Better Solutions for Complex Tasks[C]. Proceedings of the 5th International Conference on Genetic Algorithms(GA-93). Morgan Kaufmann, 1993: 264-270.
- Rosin C D, Belew R K. Methods for Competitive Co-evolution: Finding Opponents Worth Beating[C]. Proceedings of the Sixth International Conference on Genetic Algorithms. Morgan Kaufmann, 1995: 373-380.

- Miconi T. When Evolving Populations is Better than Coevolving Individuals: the Blind Mice Problem[C]. Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03), Acapulco, Mexico. MIT Press, 2003.
- Potter M A, Jong, K A D. Cooperative Co-evolution: An Architecture for Evolving Coadapted Subcomponents[J]. Evolutionary Computation, 2000, 8(1).
- Cao Y U, Fukunaga A S, Kahng A B. Cooperative Mobile Robotics: Antecedents and Directions[J]. Autonomous Robots, 1997, 4(1): 7-23.