

一种指纹特征点连线穿越脊线数的计算方法

仲伟波 吕 园 李敏敏

(江苏科技大学电信学院 镇江 212003)

摘 要 指纹中特征点间连线穿越的脊线数由于不受指纹图像缩放、旋转、平移及轻微变形的影响,被越来越多地用于指纹匹配,其计算精度直接影响指纹匹配结果。现有算法大多基于 Bresenham 及其改进方法,计算误差较大。给出了一种基于细化指纹图像的新方法,该方法首先由两特征点连线上的像素点灰度值确定其穿越的黑点坐标,然后根据连线与其相邻脊线、特征点的几何关系综合处理得到其穿越的脊线数。实验结果表明,该算法稳定可靠,能够准确地获得特征点间连线穿越的脊线数。

关键词 穿越脊线数,特征点间连线,指纹匹配,鲁棒性,精度

中图分类号 TP391 文献标识码 A

Counting Method for Ridges Crossed by Connection Line of Two Minutiae

ZHONG Wei-bo LV Yuan LI Min-min

(College of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang 212003, China)

Abstract The number of ridges crossing the connection line of two minutiae is increasingly used in fingerprint matching for its robustness in fingerprint scaling, rotation, translation and slight deformation, and affects the accuracy and robustness of fingerprint matching directly. Most ridges counting algorithms based on Bresenham or its improved algorithm are not accuracy enough now. A new ridge counting algorithm based on the thinned fingerprint was presented in this paper. The pixels of the connection line are calculated firstly, and the black pixels on or nearby the connection line are achieved. Then the number of ridges crossing the connection line is obtained according to the geometric relationship among the connection line, adjacent ridges and minutiae. Experiment results show that the algorithm can accurately and effectively get the number of ridges which across the connection line of two minutiae.

Keywords Number of ridges across connection line, Connection line of two minutiae, Fingerprint matching, Robustness, Accuracy

1 引言

指纹识别是生物识别中发展较为成熟的技术之一^[1],特征点间的距离是指纹识别的常用参数。但欧氏距离受指纹图像畸变和缩放的影响较大,而指纹缩放、畸变中两特征点连线所穿越的脊线数可保持不变,故其比欧氏距离更具鲁棒性。

两特征点连线穿越的脊线数已被广泛应用于指纹识别中。文献[2]利用特征点间连线穿越脊线数、方向差等结构关系建立局部特征向量,很好地解决了指纹平移及旋转带来的匹配问题;文献[3]把特征点连线穿越的脊线数作为判断特征点三角形是否相似的重要参数;文献[4]将特征点与参考点连线穿越的脊线数作为指纹匹配的重要判据;文献[5]也把特征点间的脊线数作为指纹拓扑模式中的一项指标,并取得了较好的匹配效果和较高的鲁棒性。

目前,计算指纹特征点间连线穿越的算法大多基于 Bresenham 及其改进算法。文献[6]以特征点间连线的像素值变化次数表示所穿越的脊线数;文献[7]利用改进的 Bresenham

算法确定两点连线上像素点坐标并判断其灰度值:若其灰度值为 0,则认为穿过一条脊线;若灰度值为 255 且在其 8 邻域内有不低于 2 个的点灰度值为 0,则认为穿过一条脊线,此算法在连线和脊线有重合时结果偏大;文献[8]统计连线上像素点中白点和黑点个数,并将其比值作为黑点间平均距离,利用黑点间平均距离解决连线与脊线重合带来的误差,由于所用黑点间平均距离并非脊线间平均距离,所得结果也不够准确。

上述计算指纹特征点间连线穿越脊线数的算法准确度、鲁棒性都不高,而特征点间连线穿越脊线数的精度直接决定着指纹匹配结果。为此,本文给出了一种新的特征点间连线穿越脊线数的计算方法,该算法基于二值化、细化的指纹图像,根据两点连线上像素点坐标、灰度值及指纹特征点坐标、脊线追踪结果,综合判断后得到其穿越的脊线数。首先根据两特征点连线斜率确定采用 x 或 y 步进获得连线上像素点坐标,然后根据连线像素点灰度值、指纹特征点坐标及脊线上像素点坐标,将属于同一脊线的像素点合并,最后得到两特征点间连线穿越的脊线数。实验表明,本文算法充分利用了指纹

本文受江苏高校优势学科建设工程项目,江苏省基础研究计划自然科学基金(BK2009623),江苏省政府留学奖学金资助。

仲伟波(1975—),主要研究方向为模式识别与智能系统、信号与信息处理,E-mail:vebopost@sohu.com;吕 园(1990—),硕士生,主要研究方向为模式识别及应用;李敏敏(1987—),硕士,主要研究方向为拓扑指纹模式构建及识别算法。

特征点坐标、脊线信息判断其穿越的脊线数,计算结果比仅判断像素点灰度值变化更准确可靠。

2 算法原理

本文采用文献[9-11]基于 Gabor 滤波的指纹图像增强。文献[12]通过自适应动态阈值二值化以及查表法细化算法进行指纹图像预处理并进行特征点提取以及伪特征点去除。利用文献[13,14]方法进行脊线追踪时采用一个 3×3 窗口作为脊线追踪前进识别器,根据当前点 8 邻域像素点的分布特征及脊线上的前驱点和当前点寻找脊线上的下一点。原始指纹及其增强、二值化、细化和脊线追踪处理的结果如图 1 所示。



图 1 原始指纹图像及其预处理的结果

算法首先在预处理后的指纹图像上获得两特征点连线的像素点坐标及灰度值,设 a, b 两特征点坐标分别为 (x_0, y_0) 、 (x_1, y_1) ,若 $ab \parallel x$ 轴或 $ab \parallel y$ 轴,则 x 或 y 步进依次判断每个像素点的灰度值,若为黑点且未被保存过,则将其按顺序存入链表,若为白点,则判断其 8 邻域点,若其中有黑点且未被保存过,则将其存入链表,依次判断直到 b 点,并将其存入链表。

若 ab 不平行 x 和 y 轴,则其连线 ab 方程为:

$$\frac{y - y_0}{x - x_0} = \frac{y_1 - y_0}{x_1 - x_0}$$

令 $\Delta X = x_1 - x_0, \Delta Y = y_1 - y_0$,若 $abs(\Delta Y) \geq abs(\Delta X)$,则 y 步进求相应 x 值,反之 x 步进求相应 y 值。由于像素点坐标都是整数,对求得的坐标值需上取整和下取整,若所得像素点是黑点且未被保存过,则将其存入链表;若得到的是 1 个白点,则需对其 8 邻域进行判断,若得到 2 个不同的白点,则需对其 10 邻域进行判断,若在其 8/10 邻域中有黑点且未被保存,则将其存入链表;依次判断直到 b 点,并将其存入链表。此处需注意 x 步进和 y 步进时的 10 邻域不同,如图 2 所示。

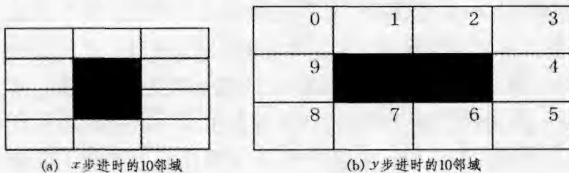


图 2 不同步进时的 10 邻域示意图

但当端点、叉点距特征点连线较近时,如图 3(a)-(c)所示,在判断连线上白点的 8/10 邻域时,端点、叉点及其相邻像

素点会被保存,致使连线穿越脊线数计算错误。为此,在对位于特征点 8 邻域内的连线白点的 8/10 邻域进行处理时,即使有黑点,但其若不在两特征点连线上则不予以保留,若位于两特征点连线上则保留;且若黑点为叉点,无论其是否在连线上都不予保留。当有与两特征点连线平行或近似平行的脊线存在时,如图 3(b)、(d)所示,在进行连线上白点 8/10 邻域搜寻时易将相邻脊线上的点计入而导致计算错误。为此在进行连线上白点 8/10 邻域搜寻得到的黑点集合若存在某些黑点在连线一侧且其连线斜率与特征点连线斜率相似,则将此黑点全部删除。

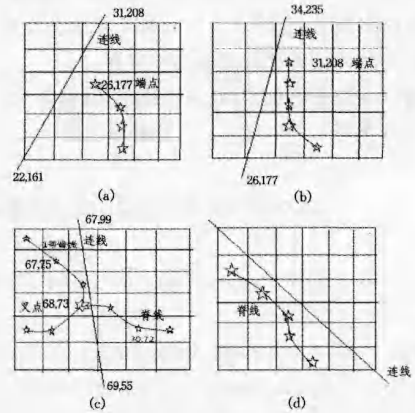


图 3 脊线与连线位置距离较近示意图

算法规定与 a 或 b 在同一脊线上的点不计,如图 4(a)所示, c 和 a 在同一脊线上,则 ab 连线穿越脊线数计为 0;如图 4(b)所示, a, b 在同一脊线上,连线 ab 穿越脊线数为 0。由于判断白点 8/10 邻域并且 b 点是连线的末端点,若去掉端点,结果会偏大,因此应将端点存入链表。



图 4 连线端点与穿越脊线数关系示意图

获得两特征点连线及其相关邻域内黑点后需根据脊线数据进一步处理,将属于同一条脊线的黑点进行合并,仅保留一个。方案一是对两特征点连线上的每一个黑点遍历所有脊线,记下其所在脊线;然后将同一脊线上的黑点进行合并,最后得到两特征点间连线穿越的脊线数。方案二遍历每一条脊线,寻找所得两特征点连线上的所有黑点,若有两个或多个黑点在同一脊线上,则仅保留一个,最后统计黑点的个数即可得到两特征点间连线穿越的脊线数。若设指纹中脊线条数为 N ,平均每条脊线上的黑点个数为 M ,两特征点连线上黑点个数为 P ,一般 $P \ll M$,经分析可知,方案一的时间复杂度较小,故采用方案一的方法。

综上所述,本文首先采用 x 或 y 步进求两特征点连线上像素点坐标,为防止漏算,对两特征点连线上白点的 8/10 邻域进行搜索,再根据指纹特征点坐标及脊线追踪数据进一步处理。计算结果比仅判断像素点灰度值变化更准确可靠。本文算法的总体流程图如图 5 所示。

纹图像中像素点的坐标为整数,而使用连线方程进行坐标计算时结果为浮点数,在进行取整时产生误差而致使穿越的脊线部分像素点在连线白点的 8/10 邻域内,虽然算法中对这些情况做了相应的处理,但最后计算结果比实际大 1。

表 1 连线穿越脊线数部分计算结果

两细节点坐标	脊线数	两细节点坐标	脊线数
(208,166)-(269,194)	6	(237,232)-(282,249)	5
(208,166)-(179,128)	4	(237,232)-(298,274)	8
(208,166)-(219,144)	1	(237,232)-(301,281)	9
(208,166)-(251,115)	2	(237,232)-(269,194)	0
(208,166)-(248,151)	1	(237,232)-(291,222)	4
(99,75)-(111,90)	2	(99,75)-(69,55)	2
(99,75)-(120,106)	5	(99,75)-(122,61)	0
(99,75)-(68,73)	1		

此外,我们在中科院自动化所的多采集交叉匹配指纹数据库的 9 个子库中各随机取 3 枚共 27 枚不同指纹进行计算,任意两特征点连线穿越脊线计算准确率都在 99% 以上。且通过分析计算结果发现,出现计算误差的原因基本上是在进行连线上白点的 8/10 邻域搜索时非穿越的相邻脊线上的点被记录,从而使计算结果出现偏差。

结束语 本文给出了一种基于细化指纹图像、指纹细节点坐标及脊线数据的两特征点间连线穿越脊线数的计算方法。实验结果证明本文给出的方法能够准确有效地获得特征点间连线穿越的脊线数。本文算法的提出使得两特征点间连线穿越脊线数这一重要参数在指纹匹配中的应用得以保证,这一参数的应用可以提高现有指纹匹配算法的准确性和鲁棒性。当然,本文算法较其他计算两特征点间连线穿越脊线数的方法复杂,且在脊线与两特征点间连线较近、斜率相似时,计算结果比实际的偏大,同时本文算法是在预处理后的细化指纹图像上进行的,并且应用了指纹特征点坐标、脊线坐标等数据信息,指纹的预处理结果和上述数据对算法结果的精确度也有重要的影响。实际实验数据表明,本文算法计算结果的准确率达到 99% 以上,能够满足实际的指纹匹配对这一参数的要求。

参考文献

- [1] Jain A K, Feng Jian-jiang, Nandakumar K. Fingerprint Matching [J]. Journal of Computer, 2010, 2: 36-44
- [2] 廖阔, 杨万麟. 点模式指纹匹配算法研究与实现[J]. 电子科技大学学报, 2004, 33(2): 154-157
- [3] 贾聪智, 解梅, 李庆嵘. 基于矢量三角法的指纹特征匹配算法的研究[J]. 计算机应用, 2004, 24(7): 45-47
- [4] 仲元昌, 朱树阔, 林永等. 一种基于 Core 点及特征点的指纹匹配算法[J]. 微电子学与计算机, 2011, 28(1): 34-37
- [5] Zhong Wei-bo, Ning Xin-bao, Wei Chen-jian. A Fingerprint Matching Algorithm Based on Relative Topological Relationship Among Minutiae[C] // IEEE Int. Conference Neural Networks & Signal Processing, June 2008; 225-228
- [6] 姜红超, 张高伟. 基于脊线跟踪的指纹细节特征提取[J]. 计算机工程, 2007, 33(8): 185-186
- [7] 苑玮琦, 李宏伟. 基于指纹结构特征信息匹配的算法[J]. 光电工程, 2006, 33(7): 101-104
- [8] 吕玉华, 李丽娟. 一种结合结构信息的指纹细节点特征提取算法

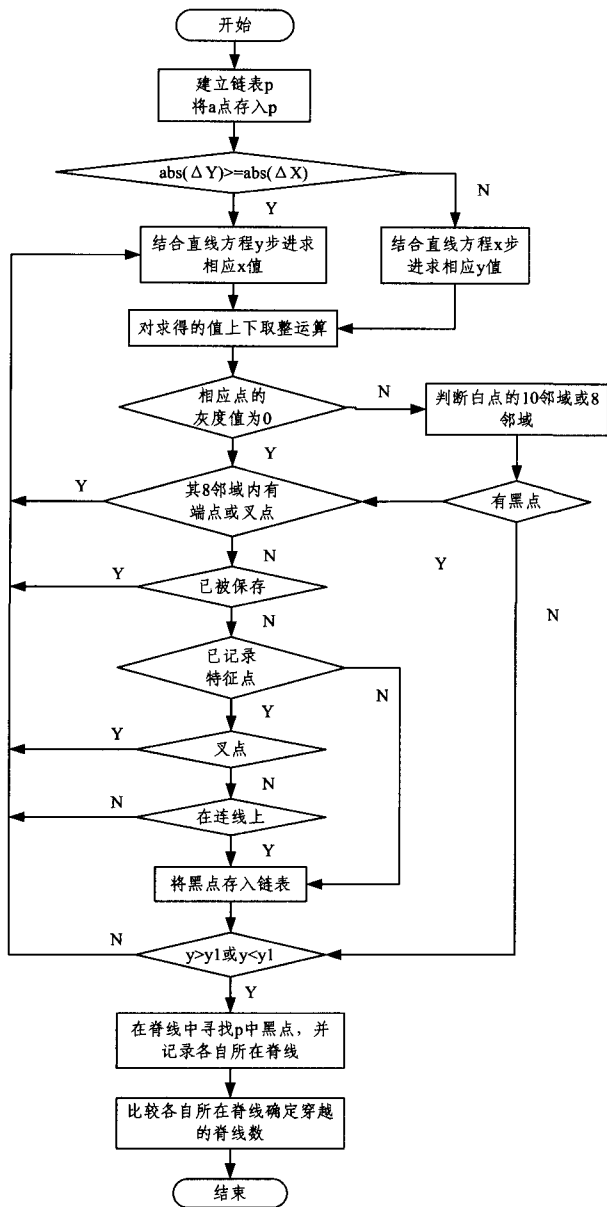


图 5 算法流程图

3 实验结果

为验证本文算法的有效性和可靠性,我们对多枚指纹图像进行了计算验证。对于如图 6 所示的原始指纹图像及其预处理后的结果,表 1 给出了 3 个特征点及其周边连线穿越脊线数的计算结果,其中需注意点(208,166)到点(179,128)穿越的脊线数是 4,因为其中一条脊线穿越了两次,此种情况在该算法中算作一条。本文算法的计算结果与图像实际情况完全符合。



图 6 实例计算指纹图像

对于图 6 中所示指纹,利用本文算法计算了 300 条特征点间连线,其中有 2 条结果与实际情况不符。原因是由于指

[J]. 计算机工程与应用, 2009, 45(7): 184-186

- [9] Wang W, Li J, Huang F, et al. Design and Implementation of Gabor Filter in Fingerprint Image Enhancement[J]. Pattern Recognition Letters, 2008, 29(3): 301-308
- [10] 李铁军, 秦伟. 自适应模板大小的 Gabor 指纹增强方法[J]. 计算机工程与应用, 2010, 46(20): 216-218
- [11] 徐婉莹, 黄新生, 刘育浩 等. 一种基于 Gabor 小波的局部特征尺度提取方法[J]. 中国图像图形学报, 2011, 16(1): 72-78

- [12] 李昊, 傅曦. 精通 VC++ 指纹模式识别系统算法及实现[M]. 北京: 人民邮电出版社, 2008: 68-133
- [13] 刘文星, 王肇析. 纹线跟踪及其在细化指纹后处理中的应用[J]. 光电子·激光, 2002, 13(2): 184-187
- [14] Ran Chong-Jie, Xie Mei. A new fingerprint matching method based on ridge tracing[C]//International Conference on Wavelet Analysis and Pattern Recognition. Beijing, China, Nov. 2007 (3): 402-407

(上接第 85 页)

线程调度仿真测试模拟系统中线程的调度执行过程, 以及各线程分派时间、运行的时间等。通过该项测试可以清楚展示调度运行是否正常、线程运行是否满足设计的周期时间要求等内容。如图 7 所示, 该系统的线程调度执行过程正常, 并且运行时间在设计的周期时间之内。

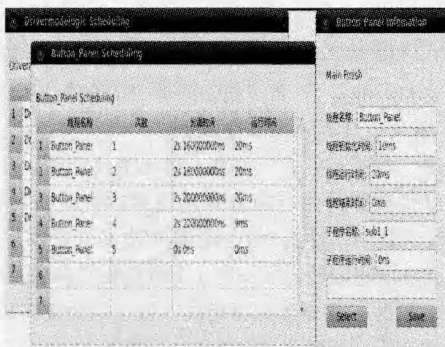


图 7 线程调度运行实时性测试数据

仿真测试还能检测系统调度执行的资源使用信息, 如线程切换次数、上下文切换次数、仿真执行指令条数、总线执行时间、中断次数、指令 cache 失效次数。如图 8 所示, 汽车控制系统仿真执行未出现异常情况, 资源使用正常。

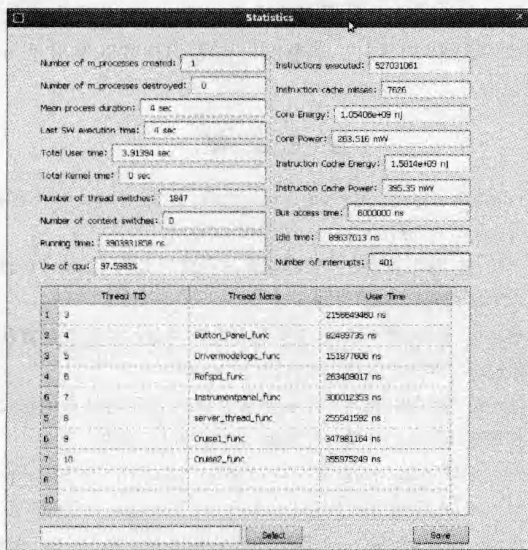


图 8 功耗和资源使用率测试数据

结束语 面向 AADL 模型测试能够动态验证 AADL 模

型的非功能属性。本文设计并实现的 AADL 模型仿真测试引擎 AMSE 可将刻画 AADL 模型实例的 System C 仿真实例文件装载在仿真测试引擎上, 通过进程封装、软硬构件绑定, 实现对系统模型的模拟运行。用户根据测试需求, 最终可得到实时性、资源使用率和功耗等方面的测试结果。本文选取了汽车巡航控制系统进行测试, 通过平台对汽车控制系统模型的测试, 详细获取了该系统的仿真测试结果, 在模型设计的早期, 可以及时验证模型的设计是否满足如实时性、CPU 使用率、功耗等非功能属性的要求, 仿真测试的结果可以为 AADL 模型优化提供依据。该仿真测试引擎对规模较大的模型驱动开发过程有较高的工程使用价值, 对模型开发早期的动态演示、模型的早期验证工作均有较大的意义。

参考文献

- [1] Feiler P H, Gluch D P, Hudak J J. The architecture analysis & design language (AADL): An introduction[R]. Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst, 2006
- [2] SAE-AS5506; SAE Architecture Analysis and Design Language (AADL). International Society of Automotive Engineers, 2005. 9
- [3] The Cheddar Project; A Free Real-Time Scheduling Analyzer [OL]. <http://beru.univ-brest.fr/~singhoff/cheddar/index.html>, 2008
- [4] Sokolsky O, Lee I, Clark D. Schedulability Analysis of AADL models[C]//IEEE International Parallel & Distributed Processing Symposium. Greece, 2006
- [5] Open Source AADL Tool Environment (OSATE) [OL]. <http://www.aadl.info/aadl/currentsite/tool/osate.html>. June, 2010
- [6] Lee S-Y, Mallet F, de Simone R. Dealing with AADL End-to-end Flow Latency with UML MARTE[C]//13th IEEE International Conference on Engineering of Complex Computer Systems. Belfast, Northern Ireland, 2008
- [7] SCoPE V1. 1. 0 [OL]. <http://www.teisa.unican.es/scope>. February, 2009
- [8] IEEE Standard for Information Technology-Portable Operating System Interface (POSIX)-Part1[S]. System Application Program Interface (API-Amendment 1; Real-time Extension
- [9] International Society of Automotive Engineers, SAEAS5506 Annex Behavior_Specification V2. 0[S]. Annex Behavior Language Compliance and Application Program Interface, USA, September, 2007