

文章编号:1001-9081(2007)01-0172-02

## 基于 ARM 的指纹特征点快速提取改进算法

游林儒<sup>1</sup>, 丁 飞<sup>1</sup>, 劳中建<sup>1</sup>, 毕淑娥<sup>2</sup>

(1. 华南理工大学 自动化科学与工程学院, 广东 广州 510640;

2. 华南理工大学 电子与信息学院, 广东 广州 510640)

(defiet@tom.com)

**摘 要:**实时性一直是嵌入式指纹识别系统开发的主要衡量标准, 如何对指纹图像进行快速、准确处理是指纹识别系统的关键所在。针对 ARM 处理平台提出一种改进的指纹特征点提取算法, 该算法的快速性和准确性已在 ARM 主建平台上得到了验证。

**关键词:**方向图; 滤波; 二值化; 细化; 特征点

**中图分类号:** TP391.41 **文献标识码:** A

## Improved fingerprint feature extraction algorithm based on ARM

YOU Lin-ru<sup>1</sup>, DING Fei<sup>1</sup>, LAO Zhong-jian<sup>1</sup>, BI Shu-e<sup>2</sup>

(1. College of Automation Science and Engineering, South China University of Technology, Guangzhou Guangdong 510640, China;

2. College of Electronic and Information Engineering, South China University of Technology, Guangzhou Guangdong 510640, China)

**Abstract:** Real-time has always been the main criterion for embedded fingerprint identification system developing. How to process the fingerprint image fast and accurately is the key to fingerprint identification system. An improved fingerprint feature extraction algorithm based on ARM was presented, and this algorithm had been proved to be fast and accurate.

**Key words:** directional image; filter; binarization; thinning; feature

### 0 引言

随着指纹识别技术发展日益成熟, 指纹身份验证系统得到越来越广泛的应用, 尤其是嵌入式指纹识别系统以自身的灵活性、方便性、易操作性得到广大用户的青睐。而对这种微型识别系统来说, 实时性是衡量系统性能一个重要标准。虽然目前指纹算法已经达到很高的水平, 但指纹的庞大数据量和算法的复杂性对微型识别系统仍是一个很大的考验。目前很多指纹匹配算法都是基于特征点, 如何快速提取特征点是微型识别系统必须解决的难题。而预处理是指纹识别中非常关键的一环, 它的好坏将直接影响到特征点提取的效果。对于预处理增强算法, 很多学者提出自己的见解。例如基于傅立叶变换的滤波算法<sup>[1]</sup>, Gabor 滤波器增强算法<sup>[2]</sup>, 都取得不错成效, 但是运算量很大, 速度慢, 不适合实时微型系统。本文基于由 Atmel 公司的 AT91FR40162 主建的 ARM 平台, 采用一种适合于 ARM 系统的、快速准确的指纹特征点提取算法, 算法包括以下步骤: 指纹图像方向图求取、滤波、二值化、细化、提取特征点。为了提高运算速度, 本文对其中的方向图求取和细化算法进行改进, 在不影响处理效果的前提下尽量避免浮点运行, 减少乘除运算, 达到快速的目的。

### 1 指纹图像的预处理

本文采用方向滤波算法对原指纹图像进行去噪、增强处理, 利用基于方向的二值化算法对处理结果进行二值化。整个过程中方向的求取起到决定性的作用。

#### 1.1 方向图的获取

采用 Veridicom 公司的 FPS200 指纹采集芯片采集到一幅

300×256 的 8 位灰度图像, 要求从图像的灰度信息中求出每点像素的方向, 方向用离散的 8 个方向值 ( $d = 1 \sim 8$ ) 表示, 每个方向夹角为  $\pi/8$  弧度。方向图就是由每点像素的方向值构成的、表征指纹方向分布的图像。

目前求取指纹方向的算法主要有预定方向逼近法<sup>[3]</sup>和 Rao 法的改进型算法<sup>[4]</sup>。前一种方法是基于 8 个方向灰度差的最佳逼近, 计算量大, 对噪声非常敏感; 后一种方法是基于梯度信息, 精度有明显提高, 但由于包含大量的三角函数运算, 仍存在速度慢的问题。文中对 Rao 法的改进型算法进行改进, 采用 sobel 梯度算子, 并结合离散分类方法求出 8 个方向值。

Rao 法首先将图像分成  $w \times w$  大小的块, 利用 sobel 算子, 计算每一个块内的每个像素点在  $x$  轴和  $y$  轴的偏微分量  $\partial_x(i, j)$  和  $\partial_y(i, j)$ 。并按下式估算每一个  $w \times w$  块的平均方向。

$$V_x(i, j) = \sum_{u=i-w/2}^{i+w/2} \sum_{v=j-w/2}^{j+w/2} 2\partial_x(u, v)\partial_y(u, v) \quad (1)$$

$$V_y(i, j) = \sum_{u=i-w/2}^{i+w/2} \sum_{v=j-w/2}^{j+w/2} [\partial_x(u, v)^2 - \partial_y(u, v)^2] \quad (2)$$

$$\theta(i, j) = \frac{1}{2} \arctan \frac{V_x(i, j)}{V_y(i, j)} \quad (3)$$

块大小的选取对图像方向的求取结果有很大的影响: 块越小, 对噪声越敏感, 受到噪声干扰越大; 块越大, 对脊线弯曲的地方, 特别是中心位置求得的方向偏差越大, 块与块之间的方向连续性受到破坏。为了解决这个矛盾的问题, 本文采取由大块图像梯度信息来求取小块方向的方法, 如图 1(b) 所示。这样既可以利用大块图像的脊线走向趋势, 减少噪声干

收稿日期: 2006-07-05; 修订日期: 2006-09-29

作者简介: 游林儒 (1956-), 男, 福建上杭人, 教授, 博士, 主要研究方向: 模式识别、智能控制、数据安全; 丁飞 (1980-), 男, 广东高州人, 硕士研究生, 主要研究方向: 智能控制、模式识别; 劳中建 (1981-), 男, 广东廉江人, 硕士研究生, 主要研究方向: 智能控制、数据安全; 毕淑娥 (1956-), 女, 黑龙江五常人, 副教授, 主要研究方向: 信号处理。

扰,又可以保持小块与小块之间的方向连续性。由试验结果可知大块选用  $13 \times 13$ ,小块选用  $7 \times 7$  时效果最好。

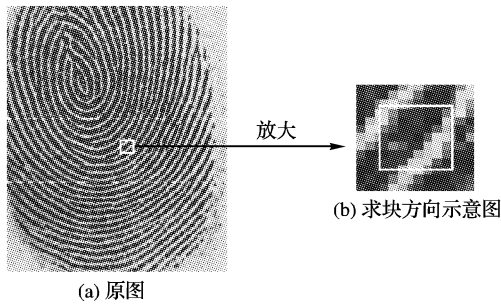


图 1 块图像的获取

在公式(3)中存在三角函数,这种浮点运算会增加程序的运算量,减缓运算速度。其实要对  $\theta(i,j)$  进行量化为 8 个方向值,可以利用反正切的对应关系直接对  $V_x(i,j)/V_y(x,y)$  的运算结果进行量化处理,同样可以达到正确的结果。因为反正切的取值范围是  $-\pi/2$  弧度到  $\pi/2$  弧度,将此区间划为 8 等份,其中  $-\pi/16$  弧度到  $\pi/16$  弧度范围的方向  $d=1$ , $\pi/16$  弧度到  $3\pi/16$  弧度范围的方向  $d=2$ ,以此类推如图 2 所示,其中括号里面为对应弧度的正切值。

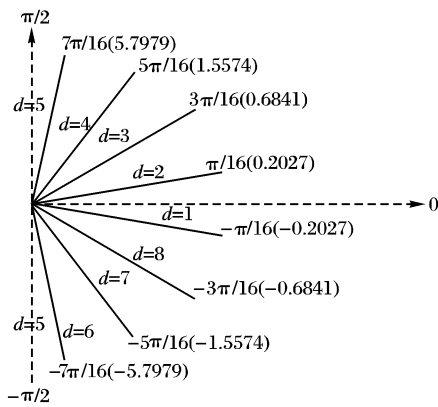


图 2 方向的量化区域分布

找出了方向的量化区域分布和正切值的对应关系,就可以利用  $V_x(i,j)/V_y(i,j)$  的计算结果,并结合  $V_x(i,j)$  和  $V_y(i,j)$  的符号来进行方向的量化了。图 3 是对图 1(a)所求的块方向图。

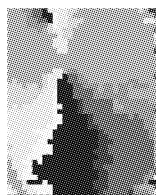


图 3 方向图

在求得的块方向图中还会有块噪声的存在,本文采用基于连续方向图的指纹智能预处理算法<sup>[5]</sup>纠正块噪声。

### 1.2 滤波和二值化

在得到指纹的方向图后,可以根据每个像素点的方向值利用方向滤波器对指纹进行滤波,以消除噪声、增强纹线、提高脊和谷之间的反差。由于指纹方向图存在八个方向,需要针对每个方向设计相应的滤波模板<sup>[6]</sup>,模板大小为  $7 \times 7$ 。滤波时,在指纹图上,从上到下,从左到右,逐点移动,根据每一个像素的方向值,选用相应的滤波器模板进行卷积滤波运算:

$$f'(x,y) = \sum_{i=-3}^3 \sum_{j=-3}^3 f(x+i,y+j)g_{\theta}(i,j) \quad (4)$$

其中  $g_{\theta}(i,j)$  为相应滤波模板的系数, $f(x+i,y+j)$  为原图像素值, $f'(x,y)$  为滤波后像素值。滤波结果见图 4。

图 4 是增强后的 256 级灰度图像,为了方便后面提取特征点,进一步将指纹图像二值化。二值化指纹图像是将灰度图像变成 0、1 两个灰度级的图像,前景点(指纹脊线)取作 0x00(黑色),背景点取作 0xFF(白色)。二值化算法采用基于

块方向图的二值化算法<sup>[7]</sup>,这种算法是对动态阈值二值化的进一步改进,并且计算量少,易于实现,图 5 是二值化结果。

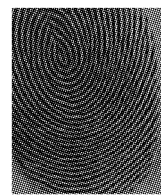


图 4 滤波结果



图 5 二值化结果

## 2 指纹图像的特征点提取

每个指纹都有各自特征点,主要是脊线端点和交叉点两种。目前许多匹配算法都是基于这两种特征点,所以准确提取特征点至关重要。对二值化图像来说,特征点难以准确定位和提取,必须进行细化,脊线只剩一个像素的宽度,有利于特征点的提取。但是细化过程耗时很大,利用 OPTA 细化算法平均要循环 10 次才能对二值化图像完全细化,在 AT91FR40162 上运行耗时 2s,这严重影响了系统的实时性。考虑到指纹特征点是随机分布在指纹图像中的,而特征点所占的面积只是整幅指纹图像的很小部分,因此对整幅图像而言,大部分区域(没有特征点的区域)是没有必要进行二值化的,只需要针对有特征点的区域进行局部细化即可,这样可以节省大部分的时间开支。

### 2.1 含指纹特征点区域的判断

在二值化图像一个特定区域内,如何判断特征点的存在呢?在指纹图像二值化后,脊线的宽度多于一个像素点。沿着脊线边界按反时针方向搜索(如图 6(a)中箭头所示方向),当搜索轨迹有一个明显的左转(如图 6(b)),表示检测到一个端点;当搜索轨迹有一个明显的右转(如图 6(c)),表示检测到一个交叉点。设  $P$  为脊线轮廓上一点, $P_{in}$  为沿搜索轨迹进入  $P$  点的方向向量, $P_{out}$  为从  $P$  点出来的方向向量。 $P_{in}$  和  $P_{out}$  由附近相邻的几个轨迹点统计计算出来,这样既可避免噪音的影响,又可以利用几个点的方向向量均值求得一个较好的方向向量。

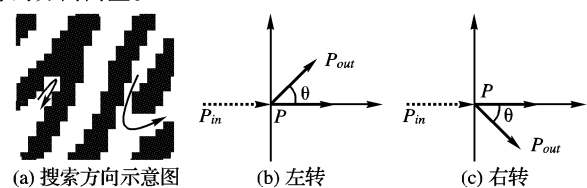


图 6 特征点搜索示意图

无论左转还是右转,都当作是特征点的存在。只要利用公式  $\theta = P_{out} - P_{in}$  在搜索轨迹上连续三个点求得方向值  $\theta_1, \theta_2, \theta_3$  满足以下两个条件,就表示该小块内含有特征点:

$$\begin{cases} |\theta_1| \geq T \\ |\theta_2| \geq T \\ |\theta_3| \geq T \end{cases} \quad (5)$$

其中: $T$  是一个给定阈值。

$$\text{sgn}(\theta_1) = \text{sgn}(\theta_2) = \text{sgn}(\theta_3) \quad (6)$$

其中: $\text{sgn}(\theta_i)$  表示  $\theta_i$  的符号( $i = 1, 2, 3$ )。

### 2.2 二值化图像的局部细化

局部细化就是对含有特征点的二值化图像某区域进行细化,将图像划分为  $25 \times 25$  的小块,采用上面提到的特征点快速判断算法对每一小块区域进行判断,有特征点即采用

(下转第 188 页)

其中 connection 是数据库连接句柄, table 是创建的表名, num\_columns 为创建的表字段个数, column\_defs 是一个用来定义表字段的 SE\_COLUMN\_DEF 结构数组, config\_keyword 是 ArcSDE 的系统关键字的字符串, 一般为 DEFAULTS<sup>[5]</sup>。这里创建的表只包含属性字段, 还未插入空间列。该函数返回 LONG 型的整数值, 根据返回值可以判断函数的执行情况。返回值为 SE\_SUCCESS, 表创建成功, 继续第 3 步。返回值为 SE\_TABLE\_EXISTS, 创建的表已经存在, 可以跳过第 3 步, 直接获得空间坐标参考系信息、图层信息等。返回值为其他时, 创建表失败, 可以根据返回值确定失败的原因。

### 3) 创建并设置图层信息

包括创建图层的空间参考坐标系、设置坐标原点及比例尺、初始化图层信息结构、设置图层的索引网格尺寸、允许的实体对象类型、创建关键字与空间列、创建图层。

### 4) 创建数据流、图形实体对象, 设置属性列与空间列

这是最为关键的一步。从数据文件读取空间对象的属性数据与空间数据, 设置相应的属性列与空间列, 依次将空间对象的信息导入数据库中。

函数 LONG SE\_stream\_set\_string(SE\_STREAM stream, SHORT column, const CHAR \* string\_val) 用于设置属性列。

其中 stream 为数据流句柄, column 为属性字段的序号, string\_val 为该字段的值。这个函数是设置类型为字符串的属性列, 其他类型的属性列选用相应的函数。

以面为例, 给出了空间列设置的主要代码:

```
//记录坐标点的数组
SE_POINT * ptrPt;
ptrPt = (SE_POINT *) new SE_POINT[ iPtNum + 1 ];
//从文件中读取坐标点的值
for (jj = 0; jj < iPtNum; jj++)
{
    fileIn >> ptX >> ptY;
    ptrPt[ jj ]. x = ptX;
    ptrPt[ jj ]. y = ptY;
}
ptrPt[ iPtNum ] = ptrPt[ 0 ];
```

(上接第 173 页)

OTPA 细化算法进行细化处理; 否则不处理。处理结果如图 7 所示。

### 2.3 特征点提取

细化后脊线只有一个像素宽度, 特征点就很容易提取出来。通过某点的八邻域像素点情况可以判断出该点是否为交叉点或端点。若为交叉点或端点, 记录下该点的坐标值, 特征点类型, 大致方向等, 这些信息用于后面的匹配需要。找到的特征点在图 7 中用一个小方框表示出来。

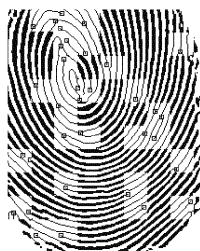


图 7 局部细化图像

## 3 结语

本文针对实时要求提出一种特征点提取的改进算法, 利用梯度算子与离散分类相结合的方法计算指纹图像块方向, 采用局部细化提取特征点的方法避免不含特征点区域的细化处理, 达到省时快速的目的。本文在以工作频率为 78MHz 的 AT91FR40162 处理芯片主建的平台上进行测试实验, 从预处理到特征点提取只需要 1s 的时间, 其中细化处理部分只需要

```
long part_offsets[ 1 ];
part_offsets[ 0 ] = 0;
//创建面对象
SE_shape_generate_polygon( iPtNum + 1, 1, part_offsets,
    ptrPt, NULL, NULL, m_seShape );
//设置空间列
SE_stream_set_shape( m_seStream, nColNum, m_seShape );
//插入数据流
SE_stream_execute( m_seStream );
delete [ ] ptrPt;
//释放对象、数据流、空间参考坐标系资源
SE_shape_free( m_seShape );
SE_stream_free( m_seStream );
SE_coordref_free( m_seCoordref );
```

## 3 结语

本次研究开发的数据入库模块, 突破了 AutoCAD DWG 数据常用入库方法的局限性, 允许用户对数据库空间表中的对象进行修改、编辑, 在数据库中创建空间表、向表中添加对象, 用户通过该模块可以设置入库条件, 具有很强的灵活性。对于本次研究的自定义格式文件或其他明码格式文件的数据入库, 本文也提供了解决的思路。

### 参考文献:

- [1] 陈继宁, 张晓东. Oracle Spatial 和 ArcSDE 的应用比较研究[ J ]. 遥感信息, 2005, (5): 60 - 63.
- [2] 陈能, 施彦琦. AutoCAD 地形图数据转换为 GIS 空间数据的技术研究与应用[ J ]. 测绘通报, 2005, (8): 11 - 14.
- [3] 韩鹏. 地理信息系统开发—MapObjects 方法[ M ]. 武汉: 武汉大学出版社, 2004. 308 - 376.
- [4] 韩勇, 陈戈, 甘宇亮, 等. 基于 CAD 数据建立 GIS 文本数据库方法探讨[ J ]. 测绘通报, 2004, (5): 22 - 24.
- [5] 芮小平, 杨建宇, 梁艳萍. 基于 ArcSDE C API 的 ArcSDE 客户端实现方法[ J ]. 物探化探计算技术, 2005, 27(4): 338 - 342.
- [6] 熊丽华, 杨峰. 基于 ArcSDE 的空间数据库技术的应用研究[ J ]. 计算机应用, 2004, 24(3): 90 - 91.

0.5s, 节省了 1.5s 的时间。但是本算法仍存在不足之处, 就是当特征点靠近小块边界或刚好在边界上, 此特征点就很容易查找不到而被忽略。如何毫无差错地查找指纹的所有特征点, 从而准确判断区域小块是否需要细化是下一步研究工作所要解决的问题。

### 参考文献:

- [1] WILLIS AJ, MYERS L. A cost-effective fingerprint recognition system for use with low-quality prints and damaged fingertips[ J ]. Pattern Recognition, 2001, 34(2): 255 - 270.
- [2] HONG L, WAN YF, JAIN A. Fingerprint image enhancement: algorithm and performance evaluation[ J ]. IEEE Transaction on Pattern Analysis and Machine Intelligence, 1998, 20(8): 777 - 789.
- [3] 韩伟红, 黄子中. 指纹自动识别系统中的预处理技术[ J ]. 计算机发展与研究. 1997, 6(34): 913 - 920.
- [4] 林征. 指纹图像方向信息的研究[ D ]. 北京: 清华大学, 1995. 25 - 28.
- [5] 曾洪波, 汪国有, 张天序. 基于连续方向图的指纹智能预处理算法[ J ]. 红外与激光工程, 2001, 30(6): 427 - 428.
- [6] 黄贤武, 王加俊, 仲兴荣. 指纹识别的预处理组合算法[ J ]. 计算机应用, 2002, 22(20): 30 - 31.
- [7] 周媛媛, 张成. 基于块方向的指纹图像预处理算法[ J ]. 图像与多媒体, 2004, (12): 57 - 59.