

# 嵌入式指纹识别系统的设计及试验研究\*

毕雪芹 苏艳娟 王 琪

(西安工业大学电信学院 西安 710021)

**摘要:** 为了降低指纹识别的拒识率和误识率,满足嵌入式指纹识别系统的实时性要求,采用基于 ARM9 处理器架构的芯片 S3C2440 的嵌入式指纹识别系统设计方案,并对二值化算法进行了研究改进。经过用本次嵌入式指纹识别系统采集指纹的试验,结果表明该方案结构简单、可扩展性强,拒识率小于 2%,较传统算法的拒识率降低 4.77%,误识率小于 1%,较传统算法的误识率降低 2.3%。本设计方案可行性高,改进的指纹图像处理算法提高了指纹图像的质量。

**关键词:** 指纹识别;算法;二值化

**中图分类号:** TN6      **文献标识码:** A      **国家标准学科分类代码:** 510.1050

## Design and experimental study of embedded fingerprint identification system

Bi Xueqin Su Yanjuan Wang Qi

(School of Telecommunication, Xi'an Technological University, Xi'an 710021, China)

**Abstract:** In order to improve the accuracy of fingerprint recognition, reduce the false rejection rate and false acceptance rate, to meet the real-time requirements of the embedded fingerprint identification system, this paper puts forward a design scheme of embedded fingerprint identification system ARM9 processor chip based on S3C2440, also study and improve two value algorithm of the fingerprint identification algorithm. Results show that, the structure of the embedded fingerprint identification system is simple, scalability and portability. The false reject rate is less than 2%, compared to the traditional algorithm; recognition rate is reduced to 4.77%. The error rate is less than 1%, compared to the traditional algorithm; the error rate is reduced by 2.3%. The design scheme has high reliability; the improved fingerprint image algorithm can improve the quality of fingerprint images.

**Keywords:** fingerprint identification; algorithm; two value algorithm

### 1 引言

多数指纹识别平台是连接 PC 的桌面应用系统,导致指纹识别系统的成本高。嵌入式平台以其处理速度快、扩展性好而受到国内外厂商的青睐<sup>[1]</sup>。嵌入式 Linux 是一种能运行于多种平台、源代码公开、免费、功能强大、与 UNIX 兼容的操作系统, Linux 自 20 世纪中期以来一直在突飞猛进地发展,目前已广泛应用于摄像机、键盘等,人们在计算机软件开发和应用中越来越离不开它,发展前景非常广阔。本文提出一种基于嵌入式 ARM9 的指纹识别系

统,设计基于 ARM9 的嵌入式指纹识别门禁系统方案,重点研究指纹图像预处理部分,采用改进的二值化算法对指纹图像进行二值化处理。

### 2 指纹识别门禁系统的硬件平台设计

以 SAMSUNG 公司 ARM9 系列 S3C2440 微处理器为硬件核心<sup>[2]</sup>,外围扩展有电源模块、时钟模块、采集模块 FPS200、存储模块、输出模块、通信模块等,搭建指纹识别门禁系统的硬件平台如图 1 所示。

收稿日期:2014-12

\* 基金项目:陕西省教育厅专项科研计划(14JK1339)资助项目

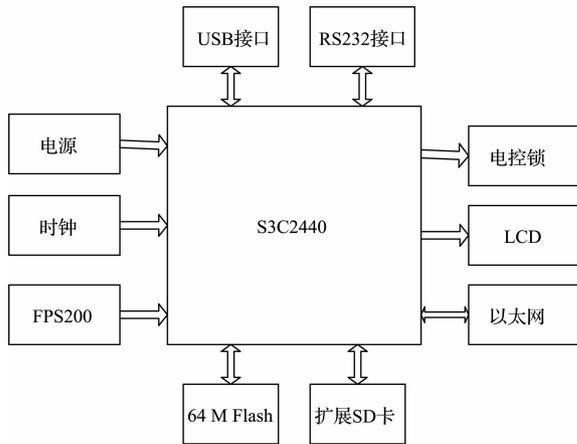


图1 指纹识别门禁系统硬件架构

### 3 系统整体软件设计方案

软件部分主要实现包括系统初始化程序、指纹采集程序以及与计算机的通信程序,控制电控锁的开关程序等。利用FPS200模块采集指纹图像,采用中断工作方式<sup>[3]</sup>,当有手指放在指纹传感器上时,系统被唤醒采集指纹图像,将指纹数据上传。图2即为该嵌入式系统指纹采集部分采得的2幅质量较好的图像,完全能满足后续指纹算法的要求。



图2 FPS200采集到的指纹图像

嵌入式Linux操作系统的开发,在主机上建立交叉编译环境,并在此环境下对内核、驱动程序及应用程序进行编译等<sup>[4]</sup>,主要有:Bootloader引导程序、Linux操作系统内核移植、指纹传感器驱动的开发、门禁系统应用程序以及其他外围应用设备等驱动的开发。本次研究采用在PC上的交叉编译环境下进行编译,生成目标板上可执行的二进制文件,通过串口下载到目标板上执行,用该方法对系统进行程序调试相当方便。

### 4 试验研究

在指纹识别过程中,指纹识别算法对指纹识别的精度有着直接影响。按指纹识别过程的主要功能,指纹识别算法可划分为图像预处理算法、特征提取算法、匹配算法<sup>[5]</sup>。

#### 4.1 指纹图像预处理

实验时,对采集的指纹图像进行处理,去除指纹采集过程中引入的各种噪声。图像预处理过程包括点方向图

计算、二值化去噪、细化。

#### 1) 点方向图的计算

采用较为简单的灰度值计算方法<sup>[5]</sup>,即依据指纹图像的灰度和灰度梯度变化计算。 $I(i, j)$ 表示原始指纹图像在点 $(i, j)$ 处的灰度值, $G(i, j)$ 就是将指纹图像灰度均值后在点 $(i, j)$ 的灰度值,而 $M$ 和 $V$ 用来代表平均值和方差,灰度值的计算公式为:

$$\begin{cases} G(i, j) = M_0 + \sqrt{\frac{V(I(i, j) - M)^2}{V}} & I(i, j) > M \\ G(i, j) = M_0 - \sqrt{\frac{V(I(i, j) - M)^2}{V}} & I(i, j) < M \end{cases} \quad (1)$$

定义一个包含多个方向的方向场,依照一个方向计算灰度梯度变化。选取一点,将灰度变化最小的方向作为这一点的方向<sup>[6]</sup>,基于嵌入式的指纹识别系统选择8个方向,图3显示了8个方向的方向场,每个方向之间的夹角为 $22.5^\circ$ 。

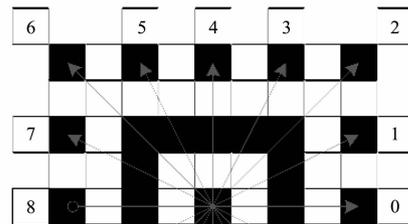


图3 灰度变化方向

设指纹中每个像素点的灰度为 $E(i, j)$ ,8个方向上的灰度改变量的总和设为 $H_d$ ,计算公式如下:

$$H_d = \sum_{k=1}^n |E(i, j) - E_d(i_k, j_k)| \quad d = 1, \dots, N \quad (2)$$

式中: $E_d(i_k, j_k)$ 表示沿着 $d$ 的方向除了点 $(i, j)$ 后的第 $k$ 个点的灰度。令 $k = 6$ ,点 $(i, j)$ 的中心顺着 $d$ 的方向两面共走 $45^\circ$ ,且 $N = d$ 即为8,然后将8个方向算出来的 $H_d$ 值进行比较, $H_d$ 最小的某个角度即为点 $(i, j)$ 的方向。

#### 2) 指纹图像的二值化

二值化是将指纹图像处理为只有0和1两个像素信息的图像。定义指纹纹线中的脊线为黑色,背景为白色。常用阈值法对指纹图像进行二值化处理,传统的阈值法是对整张图像采用一个阈值进行处理,由于同一幅图像的不同部分的灰度变化不同,这种二值化效果并不理想,得到图4。



图4 传统阈值法处理得到的二值化图像

本次采用改进后的二值化方法,将指纹图像划分为多个小块后,对每个小块采用不同的阈值进行处理,该处理算法适应性比较强,方法如下:定义每个大小为  $N \times N$  的小块阈值,计算相应的灰度均值,按照相应算法处理为 1 或 0。处理指纹灰度图像,将指纹纹线中的脊线用 1 表示,其他信息用 0 表示。如此指纹图像中,像素值为 0 的点对应灰度值为 0,像素值为 1 的点对应灰度值为 255。

每个  $N \times N$  小块中,平均灰度计算如下:

$$F(k, g) = \frac{1}{w \times w} \sum_{i=1}^w \sum_{j=1}^w G(i, j) \quad k = 1, \dots, M, \\ g = 1, \dots, N \quad (3)$$

在上述公式中,  $G(i, j)$  表示相应小块的像素点灰度值,  $w$  取 16, 而  $M, N$  的值由指纹图像的大小以及  $w$  来决定。取阈值  $T, T = F(k, g)$ , 计算小块中像素高于  $T$  的像素点个数, 记为  $N_1$ ; 低于  $T$  的像素点个数, 记为  $N_2$ 。在  $|N_2 - N_1| \leq \delta$  时 ( $\delta$  取 5),  $T$  即为所求阈值, 否则取  $T = T + 1$ , 或  $T = T - 1$ 。如果  $G(i, j) > T$ , 那么  $G(i, j) = 1$ ; 否则取  $G(i, j) = 0$ 。

改进的二值化算法处理采集到的指纹图像后, 图像就只有 0 和 1 两个像素信息, 这种处理对下一步的算法十分有利: 指纹图像的性质只由 0 和 1 两个信息决定, 相关的灰度值不用考虑。这样做会使指纹图像处理变得比较简单, 同时压缩了数据:  $256 \times 256 \times 8$  bit 的指纹灰度图像二值化后相应的数据量为  $256 \times 256 \times 1$  bit, 降低了存储成本。

### 3) 指纹图像的细化

指纹图像细化, 即在二值化指纹图像基础上, 删除多像素脊线的边缘像素, 直到脊线宽度只有一个像素<sup>[7]</sup>。细化算法采用 OPTA 算法, 通过构造消除模板和保留模板, 二值化指纹图像通过与模板的比较, 决定该点像素是否保



图5 对采集的指纹图像进行预处理

留, 此种方法在保证单像素宽度方面实现较好。图 2 采集到的指纹图像经过预处理, 得到如图 5 所示。

### 4.2 指纹特征提取

在研究了大多数的指纹特征后, 常用的指纹特征点有 2 种: 分叉点和端点。指纹特征提取算法采用 8 邻域, 使用  $3 \times 3$  的小块提取特征点<sup>[8]</sup>, 如图 6 所示。指纹端点 8 个邻域的像素只有一个是 1 的点且沿着顺时针方向, 像素值改变了 2 次; 指纹分叉点 8 个邻域的像素有 3 个像素值为 1 的点且沿着顺时针方向, 像素值改变了 6 次。

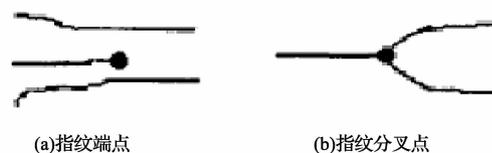


图6 常用的 2 种指纹特征点

### 4.3 指纹匹配算法

采用 FBI 提出的细节点坐标模型做细节匹配, 即通过减少被选参考点的数量来快速找到两幅指纹图像的参考点。对模板点集中的每一点和输入点集中的每一点, 定义为将和当作参照点对, 由输入图像到模板图像的旋转角度判断这两点是否是一一对应点<sup>[9]</sup>。如果是一一对应点, 即它们分别对应的脊线相似性到了一定程度, 则将取  $0^\circ \sim 360^\circ$  的一个值, 否则定义取值为 400, 以表示不能是一一对应点。如果是不同类型的细节点, 则取值为 400。

### 4.4 试验测试

从预存的指纹数据库中随机抽取了不同人的 10 个手指的指纹, 每个手指选取 10 幅图像<sup>[10]</sup>。若同一个手指的 2 幅图像没有确认就是拒识; 若不同手指的 2 幅指纹图像确认就是误识; 若相同手指确认了就是匹配。令拒识的次数为  $N_{reject}$ , 误识的次数为  $N_{error}$ , 匹配次数为  $N_{total}$ 。采取每个指纹样本与同一手指的其他指纹图像进行匹配的方法, 匹配 1 500 次, 统计拒识次数, 计算拒识率  $FRR$ 。采取每个手指样本与不同手指的每个样本进行匹配的方法, 匹配 1 500 次, 统计误识次数, 计算误识率  $FAR$ , 公式如下:

$$FRR = \frac{N_{reject}}{N_{total}} \quad FAR = \frac{N_{error}}{N_{total}} \quad (4)$$

## 5 试验结果

1) 基于以上的分析、试验, 在以 S3C2440 微处理器为核心的开发板上进行了指纹识别系统的测试, 试验结果如图 7 和 8 所示, 该指纹识别系统完成了指纹信息的采集、处理、识别过程。

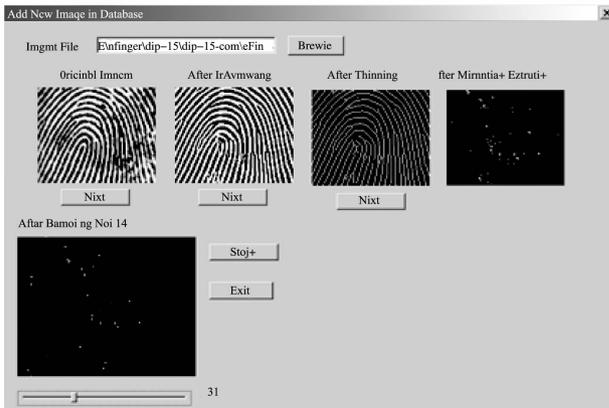
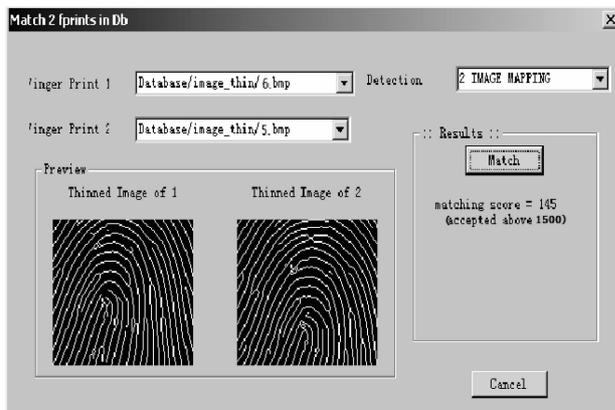
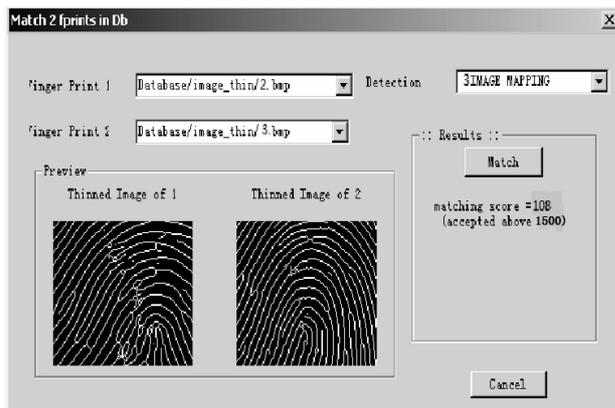


图7 指纹图像预处理与特征提取



(a)同一手指指纹图像



(b)不同手指指纹图像

图8 手指指纹图像

2)由图9与图10可以看出,二值化算法改进前,图像处理时间为272.834 ms;二值化算法改进后,图像处理时间为279.663 ms,时间误差为6.829 ms。

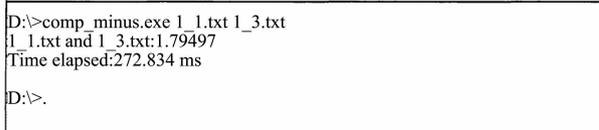


图9 优化前运行结果

```
D:\>comp_minus.exe 1_1.txt 1_3.txt
1_1.txt and 1_3.txt:1.7949708 ms
Time elapsed:272.663 ms
D:\>
```

图10 优化后运行结果

3)用传统算法和本算法经过多次匹配,得到传统算法实验结果,如表1所示;改进算法的实验结果如表2所示。

表1 传统算法的实试结果

总匹配数(次)	拒识(次)	误识(次)	拒识率	误识率	正确率
1 500	95	37	6.3%	2.5%	91.2%

表2 改进算法的实试结果

总匹配数(次)	拒识(次)	误识(次)	拒识率	误识率	正确率
1 500	23	3	1.53%	0.2%	98.27%

计算结果1: $N_{reject} = 23, N_{total} = 1 500$ 。计算得到: $FRR = 1.53%$ ,相比传统的采用固定阈值法扫描整幅图像,拒识率降低4.77%;

计算结果2: $N_{error} = 3, N_{total} = 1 500$ 。计算得到: $FAR = 0.2%$ ,相比传统的采用固定阈值法扫描整幅图像,误识率降低2.3%。

## 6 结论

采用以ARM9为核心的S3C2440处理器构建平台,连接指纹传感器FPS200,借助PC,成功采集指纹并通过串口传输到开发板上还原指纹数据。

在前人研究的基础上,对采集的指纹图像进行点方向图处理,Gabor滤波,改进的二值化算法,细化算法等进行了预处理,在此基础上利用八邻域法提取了指纹端点和分叉点,利用细节点坐标模型进行细节匹配,并将以上的算法都在软件平台上进行了实现,取得了预期的效果。

改进的指纹识别算法与传统的算法相比,程序运行时间误差很小,拒识率降低了4.77%,误识率降低了2.3%,由此证明,该方案可行性高。

## 参考文献

- [1] 王永伟,刘岩俊.嵌入式网络控制系统设计与实现[J].国外电子测量技术,2014,33(9):52-56.
- [2] 陈国金,王家校.基于ARM9的指纹识别门禁系统[J].机电工程,2009,26(9):46-50.
- [3] 于炎.指纹识别系统的需求预测[J].长春大学学报,2007,11(1):55-56.

(下转第81页)