

基于方向滤波分割的指纹自动识别系统算法

黄贤武 苏鹏程 柏培权

(苏州大学通信与电子工程系, 苏州 215021)

摘要 为适应当前个人身份鉴定的迫切需要,提出了一种基于方向滤波分割的指纹自动识别系统的完整方案。该系统主要分指纹预处理及识别两大部分,包括方向图的计算、平滑、方向滤波、二值化、去噪、特征提取、特征编码及识别的方法。利用该系统的预处理及特征编码和识别方法,指纹图象得到了较好的增强,而且,在识别过程中能克服一定范围内指纹旋转及平移因素的影响。本系统具有高效识别和很强的抗干扰能力,对于残缺或模糊的指纹图象也有很好的识别效果,有较高的使用价值。

关键词 指纹识别 特征提取 方向图 方向滤波器 二值化 去噪 指纹匹配

中图分类号: TP391.41 **文献标识码**: A **文章编号**: 1006-8961(2002)08-0829-06

Algorithms for an Automatic Fingerprint Identification System Based on Oriented Filtering and Segmentation

HUANG Xian-wu, SU Peng-cheng, BAI Pei-quan

(Dept. of Communication and Electronic Engineering, Suzhou University, Suzhou 215006)

Abstract In order to meet the instant need of individual identification, the whole project of an automatic fingerprint identification system based on oriented filtering and segmentation is presented. The project is mainly divided into two parts that are fingerprint image preprocessing and fingerprint recognition. It includes calculation and smoothness of directional image, oriented filtering, binarization, noise cleaning, feature extraction, feature coding and matching. By the methods of preprocessing, feature coding and fingerprint matching presented in this paper, the fingerprint image is effectively enhanced, moreover, during the course of fingerprint matching, certain influence of fingerprint rotation and moving can be avoided. The system has a high rate of fingerprint recognition and a strong ability to resist image noises. In addition, it is proved to be high effective in recognizing damaged or blurry fingerprint images. The system is high valuable in practice.

Keywords Fingerprint recognition, Feature extraction, Directional image, Orientation filter, Binarization, Noise cleaning, Fingerprint matching

0 概述

指纹特征是人终生不变的特征之一,而且对于不同的人,指纹特征相同的可能性几乎为零。因此,指纹识别是用于个人身份鉴定最有效的方法之一。在国内外,对于指纹识别技术已经做了不少的研究工作,但是,由于指纹的复杂性等,设计一个有效的、

实用的指纹自动识别系统仍然是一个比较困难、经久不衰的研究课题。在前人工作的基础上,根据指纹具有较强方向性的特点,提出了一套基于方向滤波分割的指纹自动识别系统的完整方案,其主要包括对指纹的预处理和识别两大部分。同时提出一种不依赖于指纹中心点的匹配算法。该指纹自动识别系统,框图如图1所示。

基金项目:江苏省自然科学基金项目(KB97035)

收稿日期:2001-04-02; 改回日期:2001-12-17

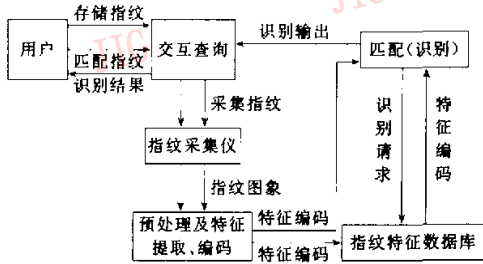


图1 系统框图

该系统具有两项功能：一是建立大容量指纹库，这一过程称为采集过程(如图 1 中虚线所示)；二是随机识别某一指纹，称为匹配过程或识别过程(如图 1 中实线所示)。在指纹采集时，摄下的指纹图象经过预处理后，把提取出的特征编码，存入数据库。在指纹匹配时，欲匹配的随机指纹图象经过预处理后，根据提取出的特征编码在指纹库中进行检索匹配，然后把查询结果返回给用户。

其中，预处理又可分为方向滤波、二值化、去噪、细化、去毛刺等^[2~4]，如图 2 所示。

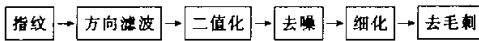


图2 预处理过程

1 指纹的预处理方法

首先采用指纹采集仪获取指纹图象，然后将获取的 16 位真彩色图象转变为 256 级灰度图象，指纹图象尺寸为 512×512 个像素。在指纹的自动识别过程中，由于受各种因素的影响，输入的指纹图象是一幅含噪音较多的灰度图象，预处理的目的是去除图象中的噪音，把它变成一幅清晰的点线图，以便于提取正确的指纹特征。预处理是指纹自动识别过程的第一步，它的好坏直接影响着指纹识别的效果。

1.1 方向滤波法

由于噪音以及压力等的影响，指纹图象获取时，将会产生两种破坏纹线：断裂及叉连。为了消除干扰及增强纹线，针对指纹纹线具有较强方向性的特点，采用方向滤波方法对其进行增强。

(1) 方向图的获取算法

方向图是用每一个像素点的方向来表示指纹图象，该系统采用了 8 个方向值，如图 3 所示。各方向之间的夹角为 $\pi/8$ ，以 $d(=1 \sim 8)$ 表示。每个像素点

上方向值的判定是在其 $N \times N$ 邻域窗口中得到的，实验中取 $N=9$ 。该 9×9 邻域窗口如图 4 所示。

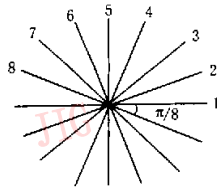


图3 8个方向

P_{71}	P_{61}	P_{51}	P_{41}	P_{31}		
P_{61}	P_{72}	P_{62}	P_{52}	P_{42}	P_{32}	P_{21}
	P_{32}			P_{11}		
P_{14}	P_{13}	$P(i, j)$	P_{12}	P_{22}		
	P_{23}		P_{33}			
P_{24}	P_{33}	P_{43}	P_{53}	P_{73}	P_{84}	
P_{34}	P_{44}	P_{54}	P_{64}	P_{74}		

图4 9×9 邻域窗口

分别求沿各个方向的灰度变化

$$S_d = \sum_{k=1}^4 |f(i, j) - f_{dk}(i_k, j_k)| \quad (1)$$

$$S_{d'} = \sum_{k=1}^4 |f(i, j) - f_{d'k}(i_k, j_k)| \quad (2)$$

其中， $d = 1 \sim 8$ ， d' 代表与 d 垂直的方向，即 $d' = (d+4) \bmod 8$ ； $f(i, j)$ 是点 $P(i, j)$ 的灰度值； P_{dk} 是 d 方向上的第 k 点； $f_{dk}(i_k, j_k)$ ， $f_{d'k}(i_k, j_k)$ 分别是点 P_{dk} 与 $P_{d'k}$ 的灰度值。

点 $P(i, j)$ 的方向应该是 S_d 取值最小， $S_{d'}$ 取值最大的方向。这不仅考虑了指纹纹线的切线方向灰度变化最小，同时也考虑了它的法线方向应是灰度变化最大的方向。当 $d=1 \sim 8$ 时，分别求出 $S_d/S_{d'}$ ，进一步得到其最小值 $S = \min(S_d/S_{d'})$ ， $P(i, j)$ 的方向取与 S 对应的 d 。

对指纹图中的每一像素，都按如上算法操作，可得到指纹图的方向图。

(2) 方向图的平滑算法

方向图求出后，还存在一定的噪音，需要对其进行平滑。

设 $N(d)$ 是某一像素 8 邻域中方向为 d 的像素的个数，定义 $N(d)$ 的最大值为 $N(D_1)$ ，次大值为 $N(D_2)$ ，其对应的方向值分别是 D_1 和 D_2 ， $C(i, j)$ 是

点 (i, j) 校正后的方向代码, 则按下式进行平滑

$$C(i, j) = \begin{cases} D_1 & \text{若 } 5 \leq N(D_1) \leq 8 \\ \frac{D_1 + D_2}{2} & \text{若 } 3 \leq N(D_1) < 5 \text{ 且 } N(D_2) \geq 2 \\ & \text{且 } N(D_1) - N(D_2) \leq 2 \\ D(i, j) & \text{其他} \end{cases} \quad (3)$$

上式中的界值由实验确定。

(3) 方向滤波器的设计

方向滤波器是由一组不同方向的多个滤波器模板组成的。先求水平方向的滤波器, 其他方向的滤波器可以通过旋转得到。一般滤波器大小为 $n \times n$, n 约为一个纹线周期。取 $n=7$, 经实验验证, 此时滤波效果最好。滤波器的系数分布如图 5 所示。

$-z/3$	$-2x/3$	$-z$	$-z$	$-z$	$-2x/3$	$-z/3$
$y/3$	$2y/3$	y	y	y	$2y/3$	$y/3$
$x/3$	$2x/3$	x	x	x	$2x/3$	$x/3$
$u/3$	$2u/3$	u	u	u	$2u/3$	$u/3$
$x/3$	$2x/3$	x	x	x	$2x/3$	$x/3$
$y/3$	$2y/3$	y	y	y	$2y/3$	$y/3$
$-z/3$	$-2x/3$	$-z$	$-z$	$-z$	$-2x/3$	$-z/3$

图 5 水平方向滤波器系数(7×7)

各系数之间按 $u > x > y \geq 0, z > 0$, 且 $u + 2x + 2y - 2z = 0$ 的关系确定。每一行的系数由中间向两端快速衰减, 其目的是为了避免破坏大曲率的纹线构型。滤波时, 指纹图中每一点的灰度值, 由其周围的 48 个点的灰度值共同决定。滤波器应有连接断点和去除叉连的作用。如果指纹中出现断点, 即这一点的灰度值比周围点小得多, 那么经过滤波器的处理后, 该点就接近周围点的灰度值了; 如果指纹中出现叉连点把相邻的两条脊线连接起来, 则其上下两行的灰度值较大, 而其同一行上邻点的灰度值较小, 经过滤波后, 叉连点的灰度值会明显降低。因此该滤波器起到了连接断点和去除叉连的作用。

得到水平方向的滤波器后, 其他方向(2~8)的滤波器可由水平方向的滤波器旋转相应的角度得到。旋转后, 滤波器上坐标 i', j' 与水平方向滤波器上坐标 i, j 之间有如下关系

$$\begin{bmatrix} i \\ j \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} i' \\ j' \end{bmatrix} \quad (4)$$

其中, $\theta = (d-1)\pi/8$ 为滤波器旋转的角度, d 为方向代码(2~8)。

在新方向的滤波器上, (i', j') 位置的系数 $g_\theta(i', j')$ 相等于水平滤波器上 (i, j) 位置的系数 $g_0(i, j)$ 。此时点 (i', j') 上的系数 $g_\theta(i', j')$ 需要用其

周围点的系数进行内插得到。设水平方向滤波器上, (i, j) 周围 4 个坐标的系数分别为: $g_0(i_L, j_L)$, $g_0(i_U, j_U)$, $g_0(i_V, j_V)$, $g_0(i_D, j_D)$, 此处 $i_L < i < i_U$, $j_L < j < j_U$, 则内插表达式为

$$g_\theta(i', j') = (j_U - j)(i_U - i)g_0(i_L, j_L) + (i - i_L)(j_U - j)g_0(i_U, j_U) + (j - j_L)(i_U - i)g_0(i_L, j_U) + (j - j_L)(i - i_L)g_0(i_U, j_U) \quad (5)$$

由此可得其余 7 个方向滤波器模板的系数, 这样 8 个滤波器模板就都得到了。

(4) 滤波

滤波时, 根据每一个像素的方向值, 选用相应的滤波器模板进行滤波(卷积运算)

$$f'(x, y) = \sum_{i=-3}^3 \sum_{j=-3}^3 f(x+i, y+j)g_\theta(i, j) \quad (6)$$

其中, $g_\theta(i, j)$ 为相应的滤波器模板的系数。

经过滤波后一些断裂和叉连现象被消除了, 纹线得到了增强。

1.2 二值化方法

以上所得的是增强后的 256 级灰度图象, 还需要将其进一步转变为二值图象(前景点(指纹脊线)取作 1, 背景点取作 0), 以把指纹脊线提取出来, 便于后续处理。根据指纹图中脊线与谷线宽度大致相等的特点, 即二值化后黑白像素的个数应大致相同。采用局部域值自适应算法, 把指纹图分成 $w \times w$ (w 约为一个纹线周期)的子块, 在每一子块内计算灰度均值

$$A = \frac{1}{w \times w} \sum_i \sum_j f(i, j) \quad (7)$$

$f(i, j)$ 为子块内 (i, j) 的灰度值。在该块内若某一点的灰度值 $f(i, j) > A$, 则 $f(i, j) = 1$, 若 $f(i, j) \leq A$, 则 $f(i, j) = 0$ 。

对每一块都进行这样的处理, 即得到二值图象。

1.3 二值化后的去噪算法

这一过程用以填补二值化后纹线上的孔洞或者删除模式上的“毛刺”和孤立的值为 1 的像素, 包括填充和删除两个过程。

(1) 填充

填充过程是把同时满足以下条件的像素 P 值取为 1:

- ① P 为 0 的像素;
 - ② P 的四邻域中有 3 个以上的邻点为 1 像素。
- (2) 删除

剔除过程是把同时满足以下条件的像素 P 值取为 0:

- ① P 为 1 的像素;
- ② $(P_1 + P_2 + P_3)(P_5 + P_6 + P_7) + (P_3 + P_4 + P_5)(P_7 + P_8 + P_1) = 0$.

其中 $P_1 \sim P_8$ (值为 0 或 1) 定义如图 6 所示.

P_4	P_3	P_2
P_5	P	P_1
P_6	P_7	P_8

图 6 3×3 模板

1.4 细化及细化后的去噪

细化就是使指纹图象的每条纹线都变为单像素宽的“点线”, 细化可去掉大量的冗余信息, 同时又能保留指纹的拓扑连接关系, 以便于后面的特征提取. 采用逐层迭代算法, 并把一次迭代分作两次扫描. 细化过程中, 由周边向中间逐层细化, 使细化结果位于原图的“中轴”.

令 N_B 为 3×3 窗口内目标像素的个数: $N_B = \sum_{i=1}^8 P_i$, 其中, P_i 定义如图 6 所示, 可取 0 或 1. N_c 为 P_8 邻域中的连接成分数, 即序列 $P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8 P_1$ 中, 0→1 变化的次数.

两次扫描中需满足的条件为

$$\left\{ \begin{array}{l} 2 \leq N_B \leq 6 \quad (\text{排除 } P \text{ 为端点和内部像点的情况}) \\ \text{若已标记 } P_i \text{ 视为 1 时, 有 } N_c = 1 \quad (\text{保证删除当前像素不会改变原图的连通性}) \\ P \text{ 的值是 1} \quad (\text{保证 } P \text{ 为前景点}) \\ \text{当 } P_3 \text{ 或 } P_5 \text{ 已标记时, 若视 } P_3, P_5 \text{ 为 0, 依然有 } N_c = 1 \quad (\text{保证宽度为 2 的线条只删除一层像点, 避免其断开}) \end{array} \right. \quad (8)$$

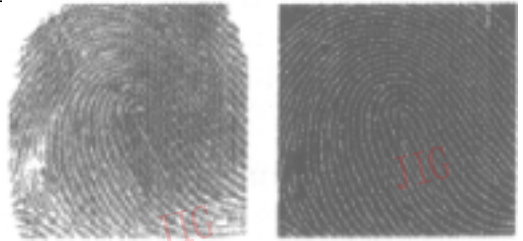
细化过程重复执行如下两个步骤(两次扫描):

(1) 从左到右, 从上到下顺序扫描图象, 对同时满足式(8)的像素(当前像素及其 3×3 邻域如图 6 所示), 如果 $P_1 P_3 P_7 = 0$ 且 $P_1 P_5 P_7 = 0$, 则将该中心像素作上标记;

(2) 从左到右, 从上到下顺序扫描图象, 对同时满足式(8)的像素, 如果 $P_1 P_3 P_5 = 0$ 且 $P_3 P_5 P_7 = 0$, 则将该中心像素作上标记.

当扫描完整幅图象后, 去掉作了标记的像素. 重复第 1、第 2 步, 直至得到单位宽度的线条为止.

经过细化过程可以得到单像素宽 8 连通的指纹图象, 如图 7 所示.



(a) 原指纹图象 (b) 处理后的指纹图象

图 7 处理前后的指纹图象

细化过程不可避免地要引入噪音, 在细化后的纹线上会出现一些毛刺, 还需要去除这些毛刺, 其方法是: 沿着纹线, 若在其垂直方向上出现长度较短的值为 1 的像素, 则将其删除.

2 特征提取及方向确定

2.1 特征提取

在指纹图中出现频度较高的细节特征有 8 种, 即点、纹线端点、分支、眼、枝叉、交叉、小桥和短线, 其中纹线端点和分支是两种基本的特征, 而其他的特征可看作由其组合而成. 取纹线端点和分支作为指纹的细节特征.

在细化后的指纹图中, 用 3×3 的模板(图 6)来进行特征提取^[5]. 先分析一下特征点(端点和分支点) 8 邻域内像素的特征. 如图 8 所示, 端点周围 8 个像素只有一个值为 1, 其余的为 0, 因此, 端点周围像素的值顺序变化(从 0→1 或从 1→0)的次数为 2; 分支点周围 8 个像素有 3 个值为 1, 其余的为 0, 因此, 分支点周围像素的值顺序变化(从 0→1 或从 1→0)的次数为 6. 根据这些特征可以判断特征点的存在.

0	0	0
0	P	1
0	0	0

0	1	0
1	P	1
0	0	0

(a) P 为端点 (b) P 为分支点

图 8 两种特征点

可以通过求交叉数 C_N 来判断细节特征, C_N 相当于某点周围 8 个像素值顺序变化次数的一半, 判断公式为

$$C_N = 0.5 \sum_{i=1}^8 |P_i - P_{i+1}| \quad (9)$$

其中, $P_i = 0$ 或 1, $P_9 = P_1$.

若 $C_N = 1$, 则模板中心为纹线端点; 若 $C_N = 3$,

则为分支点.扫描完整幅图象后,可得其全部的特征点.这些特征点绝大部分是真实的,但其中也包括由噪音引入的假特征点,因此,还需进一步根据以下规则去除假特征点:

(1) 纹线断点删除 若具有同一方向的两个端点之间的距离小于某一阈值,则认为是假特征点.

(2) 毛刺删除 若连接于分支上的端点且其到分支的距离小于某一阈值,则认为是假特征点.

(3) 位于前景区域边界的特征点应删除.

经以上步骤的处理,可删除大部分假特征点.

最后特征提取的结果如图9所示.

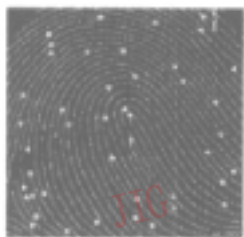


图9 特征提取结果
(特征点用白圆点标出)

2.2 端点及分支点的方向

如果仅依靠分支点和端点的位置,是无法体现出纹线的走向的.由于在一局部区域内的指纹纹线的走向大致是相同的,所以若能在分支点和端点上提取出反映纹线走向的方向,则特征点及其方向就能很好地描述指纹.提取出指纹的特征点之后,进一步求出每个特征点的方向,从而可以据此进一步求出其相互取向关系.特征点的方向定义如下^[6]:

端点的方向 设 p 为一纹线端点,从 p 出发沿脊线移动一段距离后,到达像素点 p' ,则规定从 p' 到 p 的方向为端点的方向.

分支点的方向 设分支点的中心为 p ,从 p 出发沿3条分支所在的脊线向外移动一段距离后,分别到达3个像素点 $p_1, p_2,$ 和 p_3 ,在这3个点构成的3角形中,短边所对的顶点为 p' ,则定义从 p 到 p' 的方向为分支点 p 的方向.

按以上定义可分别求出各特征点的方向.

3 特征编码及匹配算法

采用一种不依赖于指纹中心点的编码方法对特征点进行编码.这种编码方法需要用到特征点之间的相互位置关系^[4].设一幅指纹图的特征集合为 S ,

$S = \{p_1, p_2, \dots, p_n\}$, p_i 为某一特征点.用一个 n 维数组 $F[n]$ 来描述 S 中每一个元素的方向,用 n 维数组 $T[n]$ 描述 S 中元素的类型.若 p_i 是端点,则 $T[i] = 0$;若 p_i 是分支点,则 $T[i] = 1$.

为描述某一特征点 p_i ,定义一个结构 PtRelation 如下:

```
struct PtRelation {
    int type;
    float angle1;
    int dist1;
    float angle2;
    int dist2;
    float RelaAngle;
};
```

设特征点 p_i 是同一指纹中与 p_i 距离最近的端点,特征点 p_{i_2} 是同一指纹中与 p_i 距离最近的分支点.该结构 PtRelation 中, type 为该特征点 p_i 的类型; angle1, dist1 分别为该特征点 p_i 与特征点 p_{i_1} 的夹角 ($|F[i] - F[i_1]|$) 和距离 (跨越的脊线数); angle2, dist2 分别为该特征点 p_i 与特征点 p_{i_2} 的夹角 ($|F[i] - F[i_2]|$) 和距离 (跨越的脊线数); RelaAngle 为通过 p_i 和 p_{i_1} 的直线与通过 p_i 和 p_{i_2} 的直线所成的夹角.这几个变量 (angle1, dist1, angle2, dist2, RelaAngle) 反映了特征点之间的拓扑关系,描述了特征点 p_i 的邻域特性.

用数组 $PR[n]$ 来描述 S 中元素的几何相互关系, $PR[n]$ 的每一个元素属于 PtRelation 结构类型,描述相应特征点的结构信息.将 n 与 $PR[n]$ 存在一个二进制文件中,该文件就是指纹的特征编码,即:

$n, PR[0], PR[1], PR[2], \dots, PR[n-1]$

在建立指纹特征数据库时,对每一个输入的指纹图象,将处理后得到的特征点按此方法进行编码,并且保存到数据库中.

设新输入指纹图象的特征集合为 S' , $S' = \{p'_1, p'_2, \dots, p'_m\}$, p'_i 为某一特征点.用一个 m 维数组 $F'[m]$ 来描述 S' 中每一个元素的方向,用 m 维数组 $T'[m]$ 描述 S' 中元素的类型,若 p'_i 是端点,则 $T'[i] = 0$;若 p'_i 是分支点,则 $T'[i] = 1$.用数组 $PR'[m]$ 来描述 S' 中元素的几何相互关系, $PR'[m]$ 的每一个元素是 PtRelation 结构类型的,用于描述某一特征点的结构信息.

在特征匹配时,依次读取数据库中的指纹特征编码文件,并按如下方法与该新输入指纹的特征编码进行匹配,以找到与它相匹配的指纹.

对于数据库中某指纹特征编码,若 $PR[n]$ 中的某元素 $PR[i]$ 与 $PR'[m]$ 中的某元素 $PR'[j]$, 满足

$$\begin{aligned} PR[i].type &= PR'[j].type, \\ |PR[i].angle1-PR'[j].angle1| &< H_1, \\ |PR[i].dist1-PR'[j].dist1| &< H_2, \\ |PR[i].angle2-PR'[j].angle2| &< H_3, \\ |PR[i].dist2-PR'[j].dist2| &< H_4, \\ |PR[i].RelaAngle-PR'[j].RelaAngle| &< H_5 \end{aligned}$$

其中, $H_1 \sim H_5$ 为由实验确定的阈值, 则说明这两个元素的邻域特性相近, 认为 $PR[i]$ 与 $PR'[j]$ 匹配. 上述各式允许各个变量有一定的误差存在, 这就提高了该算法的抗噪音性能.

从本质上讲, 这种不依赖中心点的方法就是, 如果能从标准指纹特征集合 S 中选取一个子集 S_1 , 从新输入的指纹特征集合 S' 中选取一个子集 S'_1 (这两个集合大小相等, 即 $|S_1| = |S'_1|$), 且 S_1 到 S'_1 存在一个一一映射 g , 使 $S'_1 = g(S_1)$, 即 S_1 中的特征点与 S'_1 中的特征点是一一相互匹配的, 则认为这两枚指纹所能匹配的特征点数是 $S^* = |S_1| = |S'_1|$.

由于在编码时, 利用了每个特征点与其周围特征点之间的相互位置关系, 而不是每个特征点的绝对位置, 这样就可以克服指纹旋转所带来的影响, 而且对指纹的扭曲变形也有一定的适应能力. 编码反映了指纹的拓扑结构, 可以不必寻找其中心点来作为参考位置进行编码, 这就提高了该算法的通用性和可靠性, 因为不是每个指纹都有中心点, 所以对于这种情况就很难确定其参考位置, 而且, 即使指纹存在着中心点, 但要精确地找出其中心点也是相当困难的, 而在识别时, 利用两枚指纹特征集合中相对应的子集来进行匹配, 对于残缺或部分模糊的指纹也可以进行有效的识别.

怎样衡量两枚指纹的相似程度呢? 这应该由两枚指纹的特征点数目以及它们所能匹配的特征点数目共同确定. 为了定量描述两枚指纹的相似程度, 定义可信度为

$$C_v = \sqrt{\frac{|S^*|}{m \cdot n}} \quad (10)$$

其中, m, n 分别为两枚指纹的特征点数目, S^* 为它们所能匹配的特征点数目. 则 C_v 越大, S^* 就越接近于 m, n , 这两枚指纹就越相似. 若 C_v 大于某一阈值 T (T 值通过实验确定), 则认为这两枚指纹是相同的, 否则认为不匹配.

4 实验结果及讨论

对 50 对指纹 (其中有 10 枚指纹较模糊, 10 枚指纹是部分残缺的) 进行了 50×50 次的交叉识别实验, 实验结果为: 拒识率为 2%, 误识率小于 3%, 识别率在 95% 以上, 较好地达到了使用的要求. 该算法对指纹的旋转与形变有较好的适应能力, 对部分残缺或模糊的指纹也有较高的识别能力. 不过, 本系统也存在一些不足之处, 其中, 预处理部分需要进行较多的运算, 时间较长 (在 P III 450 的 PC 机上约 13s), 而且很多中间数据需要保存, 要占用一定的内存空间, 这些还需进一步解决.

参考文献

- 1 Andrew K Hrechak, James A Mchugh. Automated fingerprint recognition using structural matching[J]. Pattern Recognition, 1990, 23(8): 893~904.
- 2 张建伟, 陆亨立. 指纹自动识别中图象分割方法的研究[J]. 微型电脑应用, 1999, 15(12): 20~22.
- 3 Mehre B M. Segmentation of fingerprint images using the directional images[J]. Pattern Recognition, 1987, 20(4): 429~435.
- 4 XIAO Qinghan, Hazem Raafat. Fingerprint image postprocessing: A combined statistical and structural approach [J]. Pattern Recognition, 1991, 24(10): 985~989.
- 5 Nalini K Ratha, CHEN Shaoyun, Anil K Jain. Adaptive flow orientation-based feature extraction in fingerprint images [J]. Pattern Recognition, 1995, 28(11): 1659~1663.
- 6 胡骏华. 指纹锁指纹鉴定算法的研究[D]. 中国科学院自动化研究所, 1995.



黄贤武 1941 年生, 教授, 博士生导师. 主要研究方向为数字图象处理、数据压缩编码技术等. 出版著作 6 本, 发表论文 15 篇.



苏鹏程 1977 年生, 博士研究生. 主要研究方向为数字图象处理、图象通信等.



柏培权 1971 年生, 硕士研究生. 主要研究方向为数字图象处理、模式识别等.