

中图法分类号: TP391.41 文献标志码: A 文章编号: 1006-8961(2010)11-1608-07

检索信息: 王峰, 陈蕴. 连续方向图法在残留指纹分割中的应用[J]. 中国图象图形学报, 2010, 15(11): 1608-1614

连续方向图法在残留指纹分割中的应用

王峰^{1),2)}, 陈蕴¹⁾

¹⁾(阜阳师范学院计算机与信息学院, 阜阳 236041) ²⁾(山东大学计算机科学与技术学院, 济南 250101)

摘要: 针对残留指纹不易被分割的特点, 提出一种连续方向图实现残留指纹分割的应用算法。该算法首先计算点方向图, 然后利用直方图滤波技术对指纹点方向图进行滤波处理, 得到连续分布方向图; 其次对前景区域进一步分析, 根据残留指纹和正常指纹方向一般不一致的特点, 运用残留指纹与指纹区交界处方向信息变化较大的特性标记出残留指纹和指纹区边界点, 然后对标记的一些虚假点进行后处理, 最后利用这些边界点进行曲线拟合后去除残留指纹。实验结果表明这种连续分布方向图过渡平滑、自然, 既具有很好的连续性、渐变性和抗噪性, 又有较高的精度, 可以有效地分割残留指纹。

关键词: 指纹图像分割; 点方向图; 残留指纹; 连续方向图

The application of continuously distributed directional image in remaining fingerprint segmentation

WANG Feng^{1),2)}, CHEN Yun¹⁾

¹⁾(School of Computer and Information of Fuyang Normal College, Fuyang 236041)

²⁾(School of Computer Science and Technology of Shandong University, Jinan 250101)

Abstract: For the problem that the remaining fingerprint cannot be segmented easily, the authors propose a segmentation algorithm for fingerprint images with remaining ridges based on continuously distributed directional image. The algorithm firstly calculates the point directional image, and uses the histogram filtering technology to filter directional image, and then the continuously distributed directional image can be obtained. Secondly, according to the direction inconsistency on the location where the remaining ridges border on the fingerprints in the foreground, we mark out the boundary between the remaining fingerprint and normal ridge based on the sharp directional information change. After dealing with false points in the post-processing, these true boundary points are used for curve fitting. Finally we remove the remaining fingerprint. Experiments have shown that such a continuously distributed directional image transits smoothly and naturally. It not only has good continuity, gradual change and noise immunity, but also has very high accuracy. And it can effectively segment the remaining fingerprint from the foreground.

Keywords: fingerprint segmentation; point directional; remaining fingerprint; continuously directional image

0 引言

指纹分割是指纹识别系统预处理中的关键一步。指纹图像分为前景和背景两部分, 前景是指图

像中清晰的指纹部分, 而背景指的是采集区域中无纹线和纹线非常模糊的部分。指纹图像分割的目的就是要将有效的指纹区域(即前景区)同背景区域以及质量很差、在后续处理中很难恢复的模糊区域分离开来, 使后续处理能够集中于有效指纹区域, 以

基金项目: 安徽省教育厅自然科学基金项目(KJ2009B136); 安徽省高校青年资助计划项目(2007jq1145)。

收稿日期: 2009-04-07; 改回日期: 2009-07-16

第一作者简介: 王峰(1973—), 男, 副教授。2003年于安徽理工大学获计算机应用专业工学硕士学位。主要研究方向为图像处理和模式识别。E-mail: wfl11625@163.com。

避免在噪声和背景区域中提取特征,同时也利于提高特征提取的准确性,减少提取到的伪特征点,从而提高整个系统性能。

现有的指纹图像分割方法主要包括以下两类:一类是基于块水平的分割方法^[1-2],另一类是基于像素水平的方法^[3-4]。它们都是利用基于指纹图像固有的特征参数来进行分割的,如灰阶统计特征、纹线方向、投影信号等。这些方法可以有效地去除非脊线区域和不可恢复的脊线区域。然而,这些方法都没有考虑去除残留指纹区域(图 1),残留指纹是在之前采集器留下的,因为残留指纹往往具有清晰的纹路和可恢复的脊线结构,因此在分割的时候常常作为前景而导致分割失败。

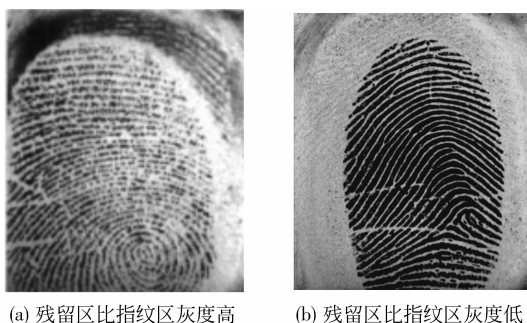


图 1 含有残留指纹的指纹图像

Fig. 1 Fingerprint image including remaining ridges

文献[5]中利用了真正脊线和残留脊线的两个区别:一是残留脊线块的像素灰度值通常比真正的脊线块大;二是残留脊线块脊谷间的灰度差别比真正脊线块小。但这两点区别并不是残留脊线固有的特征,很多情况下在同一幅指纹图像中两者的灰度值和脊谷间的灰度差别并不大,如图 1 所示,文献[6]中的分割方法不适用这种类型的残留指纹。

残留脊线与真正脊线区域的位置关系可以分为以下两类^[7]:

1) 很多情况下,残留脊线与真正脊线区域是分离的,残留指纹往往容易形成一个或多个孤立区域,

而这些孤立区域面积一般比指纹区域小得多。

2) 正常情况下采集的指纹图像边界呈凹凸性,当残留脊线与真正脊线区域粘连时,残留指纹图像边界的形状相对异常(异常指的是边界图像的像素点的 8 邻域有带有纹理的像素点存在,如果没有残留指纹正常脊线的边界点 8 邻域应该是背景的像素点),理想的分割点应出现在凹角处,本文算法针对的就是这样一种情况。

因此,问题的关键就转化为如何求出残留指纹与正常指纹的边界点,针对残留指纹往往具有清晰的纹路而不易被分割的特点,提出一种连续方向图法在残留指纹分割中的应用算法。该算法包括两部分:一是利用直方图滤波技术对指纹点方向图进行滤波处理,得到连续分布方向图。实践证明这种连续分布方向图过渡平滑、自然,既具有很好的连续性、渐变性和抗噪性,又具有极高的精确度。二是对前景区域进一步分析,根据残留指纹和正常指纹方向一般不一致的特点,运用方向信息变化较大的特性标记出残留指纹和指纹区边界点。为验证该算法的性能,将改进算法与文献[7]算法进行了对比实验。实验结果证明了本文算法的有效性。

1 指纹连续方向图算法

1.1 指纹图像的背景分割

1.1.1 归一化

受采集设备工作环境及参数设置的影响,采集到的指纹图像可能总体对比度较差。灰度归一化的目的是将指纹图像的灰度变化调整到一个统一的范围,使不同图像具有相同的灰度均值和方差,从而使后续处理具有相同的基准。令 I 表示原始指纹图像,其大小为 $M \times N$, $I(x, y)$ 表示图像中像素点 (x, y) 的灰度值。令 G 表示灰度规格化后的指纹图像, $G(x, y)$ 为规格化后图像中像素点 (x, y) 的灰度,对输入的原指纹图像作归一化处理^[8]:

$$G(i, j) = \begin{cases} \text{Mean}_0 + \frac{\sqrt{\text{Var}_0 \times (I(i, j) - \text{Mean})^2}}{\text{Var}} & I(i, j) > \text{Mean} \\ \text{Mean}_0 - \frac{\sqrt{\text{Var}_0 \times (I(i, j) - \text{Mean})^2}}{\text{Var}} & \text{其他} \end{cases} \quad (1)$$

式中,

$$\text{Mean} = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} I(i, j) \quad \text{Var} = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (I(i, j) - \text{Mean})^2 \quad (2)$$

$Mean$ 和 Var 分别是原始图像的灰度均值和方差, $Mean_0$ 和 Var_0 分别是期望的灰度均值和方差。在此 $Mean_0$ 和 Var_0 均取 150。灰度归一化并没有改变指纹纹线的质量, 只是改善了整幅图像的灰度对比度。

1.1.2 背景分割

分割旨在割除正常指纹以外的区域。因此如果指纹图像中含有残留纹路区域, 在方向计算正确的情况下, 这些残留纹路区域很可能在初分割中被分为前景块。孤立地看残留纹路区域, 这类区域具有纹路结构, 因此一般算法难以通过一次处理将残留纹割除。

有效指纹图像区域具有良好的纹理特征^[9], 背景区域作为相对平滑区域则不具有良好的纹理特征。相对于背景区域而言, 有效指纹图像区域具有更好的方向选择性, 表现出良好的各向异性, 而背景区域则在相关性质上表现出各向同性。充分考虑有效指纹区域在其切线方向和法线方向上表现出的不同性质, 根据有效指纹区域良好的方向选择性, 分别求取 4 个不同的分割指标组成像素的特征向量, 然后采用感知器算法实现对有效指纹图像区域的分割处理。该方法相对于无方向判别方法具有更高的分割精度, 分割结果抗噪声干扰的能力更强。本文第一步分割算法正是利用有效指纹区域和背景区域在纹理表现上的不同对其进行分割处理。

1.2 计算点方向图

把指纹脊线的走向分为 8 个方向^[6], 如图 2 和图 3 所示, 基准点位于方向模板的中心, 从水平位置开始, 按逆时针方向, 每隔 $\pi/8$ 确定一个方向, 用数字 0, 1, 2, ..., 7 来表示, 此方法计算的方向角范围是 $[0, \pi)$ 。

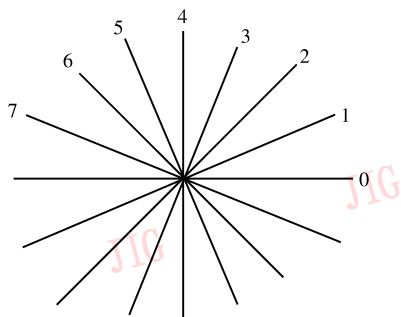


图 2 在每个像素处的 8 个方向
Fig. 2 Eight directions of each pixel

6		5		4		3		2
7		6	5	4	3	2		1
		7				1		
0		0		0		0		0
		1				7		
1		2	3	4	5	6		7
2		3		4		5		6

图 3 8 个方向数的 9×9 方向模板

Fig. 3 Eight directional 9×9 template

具体计算步骤如下:

1) 对指纹图像中的每一点 $p(x, y)$, 在以该点为中心的 9×9 窗口内, 分别计算 8 个方向上的灰度平均值, 即对图 2 中标有 i ($i = 0, 1, \dots, 7$ 分别代表 8 个方向的值) 位置的像素灰度求平均值, 得到 $G_{\text{mean}}[i]$;

2) 将这 8 个平均值按两两垂直的方向分成 4 组: 0 和 4 为一组, 1 和 5 为一组, 2 和 6 为一组, 3 和 7 为一组, 分别计算每组中两个平均值差的绝对值

$$G_{\text{diff}}[j] = |G_{\text{mean}}[j] - G_{\text{mean}}[j + 4]| \quad (3)$$

式中, $j = 0, 1, 2, 3$ 为脊线方向;

3) 取差值绝对值最大的两个方向作为可能的脊线方向, 即若

$$i_{\text{max}} = \text{ord}_{\text{max}}(G_{\text{diff}}(i)) \quad i = 0, 1, 2, 3 \quad (4)$$

则方向 i_{max} 和 $i_{\text{max}} + 4$ 为像素 $p(x, y)$ 处可能的脊线方向;

4) 取 i_{max} 和 $i_{\text{max}} + 4$ 两个方向中灰度平均值与该像素 $p(x, y)$ 的灰度值 G 比较接近的方向作为该像素点的脊线方向 $i(x, y)$, 分别对指纹图像中每一个像素点进行处理, 便可得到指纹的点方向图。

$$D(x, y) = \begin{cases} i_{\text{max}} & |G - G_{\text{mean}}[i_{\text{max}}]| < \\ & |G - G_{\text{mean}}[i_{\text{max}} + 4]| \\ i_{\text{max}} + 4 & \text{其他} \end{cases} \quad (5)$$

1.3 计算指纹的连续分布方向图

点方向图是连续变化的, 具有连续性和渐变性, 相邻像素点的方向过渡平滑自然, 但由于原指纹图像含有大量的随机噪声, 造成点方向图对噪声的鲁棒性差, 不能完全反映指纹脊线的真实走向, 因此不能直接使用。

为了克服上述缺点, 本文在计算出点方向图后, 再采用平滑滤波技术, 分别对每一个像素点采用连续滑动的 $w \times w$ 窗口 (模板) 来对点方向图 $i(x, y)$ 进行平滑处理, ($w = 17$, w 的取值为一个脊一谷的宽度

之和,由指纹脊线的周期决定),即在以此点为中心的 $w \times w$ 大小的窗口范围内进行方向直方图统计,具体做法如下:

1) 分别统计该 $w \times w$ 大小的窗口内方向值为 $i(i = 0, 1, \dots, 7)$ 的像素个数;

2) 把方向直方图中峰值所对应的方向(即 $w \times w$ 大小的窗口内方向统计数最大的一个方向)作为点 $p(x, y)$ 的方向,即

$$O(x, y) = \text{ord}(\max(N_i), i = 0, 1, \dots, 7) \quad (6)$$

式中, $\text{ord}()$ 为数组 A 的元素 A_i 的下表 i 的函数,即 $\text{ord}(A_i) = i$ 。

以此方法分别对点方向图 $D(x, y)$ 中的每一点进行平滑处理,便可得到平滑的点方向图 $O(x, y)$,实验结果证明,该点方向图抗噪声能力很强,如图 4 所示为带残留指纹点方向图平滑前和平滑后结果对比。

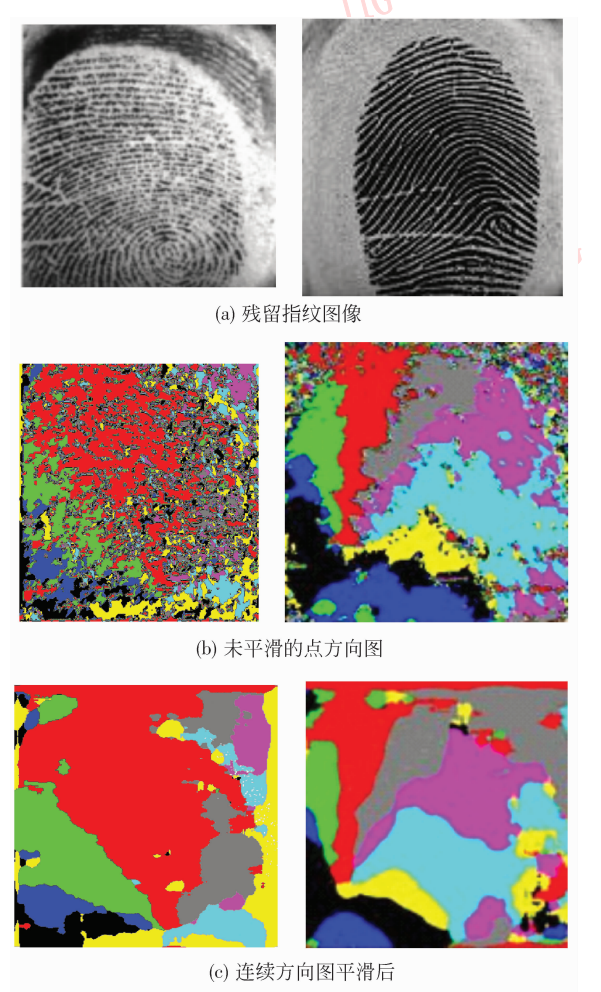


图 4 不同质量的残留指纹点方向图平滑前后的对比
Fig. 4 Different quality remaining fingerprint point directional image before and after smoothing contrast

2 基于连续分布方向图的残留指纹分割

2.1 方向变化较大区域判断

在前景区部分,有残留指纹和正常指纹,如何区别呢?利用刚刚求出的连续分布方向图的纹理性质来判断:检测指纹图像的每一个点的 8 邻域方向,当遇到点的方向与其他点的方向变化很大时,标记出可疑点,整个循环下来,把所有点遍历一遍,标记出所有可疑点。

2.2 后处理

2.2.1 错误点产生的原因

模式区是指纹上包括了总体特征的区域,它是由典型线围绕而成的指纹区域。典型线是指指纹中形成圆形或环绕趋势的最外层纹线。在模式区中,有两类特殊的点:一类叫做三角点(delta 点),另一类叫做核心点(core 点)。核心点被定义为纹线上曲率最大的点,位于指纹纹路的渐进中心,周围纹线大致半圆趋向。三角点被定义为模式区内离典型线分叉处最近的点。

针对指纹的纹理特点,在指纹的模式区域纹线方向变化迅速、曲率较大、一致性较差,根据点方向图一致性检测可疑点,使得部分模式区域的点被检测出来(如图 5 所示)。

2.2.2 消除模式区的误检点

假设将指纹图像的方向场定义在 $[-\pi/2, \pi/2]$ 上,就会发现模式(奇异点)区域点周围的方向场具有如下特性:

1) 围绕奇异点区域附近的封闭曲线上的点,其方向角的变化是连续的,即封闭曲线上点的方向由连续的正方向和连续的负方向两部分组成,若按顺时针方向遍历,该封闭曲线上点的方向由正方向变为负方向和由负方向变为正方向各有一次且仅有一次。

2) 这两次角度的变化有一次变化较大有一次变化较小,变化较小的一次是发生在 0° 附近而变化较大的一次是发生在 $\pi/2$ 或者说 $-\pi/2$ 附近,变化较大的这一次其方向角的差值随脊线曲率的大小而改变,且这次方向的突变发生在纹线结构的固定位置。

若指纹的方向场是定义在 $[0, \pi)$ 上,在指纹的奇异点区域的可疑点其周围方向角存在略大于零,和接近 π 的相邻角突变,利用这些特性去除奇异点区域的部分虚假可疑点(如图 5 所示)。

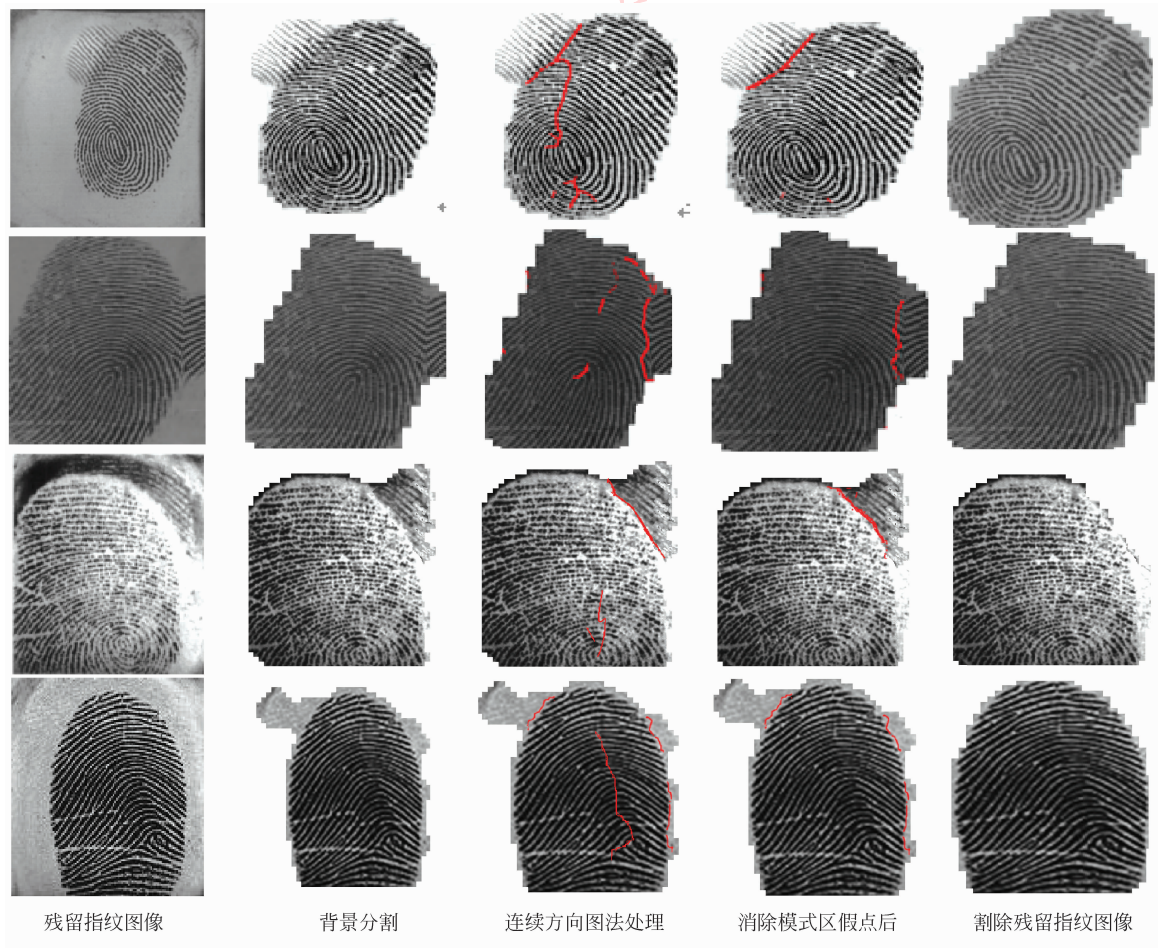


图 5 本文算法处理残留指纹结果
Fig. 5 Segmentation result for remaining fingerprint by the proposed algorithm

2.3 曲线拟合,与原来轮廓的边界对接

2.3.1 最小二乘法的基本原理

从整体上考虑近似函数 $p(x)$ 同所给数据点 $(x_i, y_i) (i=0, 1, \cdots, m)$ 逼近程度。数据拟合的具体作法是:对给定数据 $(x_i, y_i) (i=0, 1, \cdots, m)$, 在取定的函数类 Ω 中,求 $p(x) \in \Omega$, 使误差 $r_i = p(x_i) - y_i$ $(i=0, 1, \cdots, m)$ 的平方和最小,即

$$\sum_{i=0}^m r_i^2 = \sum_{i=0}^m [p(x, y) - y_i]^2 = \min \quad (7)$$

从几何意义上讲,就是寻求与给定点 $(x_i, y_i) (i=0, 1, \cdots, m)$ 的距离平方和为最小的曲线 $y = p(x)$ 。函数 $p(x)$ 称为拟合函数或最小二乘解,求拟合函数 $p(x)$ 的方法称为曲线拟合的最小二乘法。

2.3.2 多项式拟合的过程

假设给定数据点 $(x_i, y_i) (i=0, 1, \cdots, m)$, Φ 为所有次数不超过 $n (n \leq m)$ 的多项式构成的函数类,

现求 $p_n(x) = \sum_{k=0}^n a_k x^k \in \Phi$, 使得

$$I = \sum_{i=0}^m [p_n(x_i) - y_i]^2 = \sum_{i=0}^m \left(\sum_{k=0}^n a_k x_i^k - y_i \right)^2 = \min \quad (8)$$

为 $a_0, a_1, \cdots, a_n$ 的多元函数,因此上述问题即为求 $I = I(a_0, a_1, \cdots, a_n)$ 的极值问题。由多元函数求极值的必要条件,得

$$\sum_{k=0}^n \left(\sum_{i=0}^m x_i^{j+k} \right) a_k = \sum_{i=0}^m x_i^j y_i \quad j = 0, 1, \cdots, n \quad (9)$$

式(9)是关于 $a_0, a_1, \cdots, a_n$ 的线性方程组,用矩阵表示为

$$\begin{bmatrix} m+1 & \sum_{i=0}^m x_i & \cdots & \sum_{i=0}^m x_i^n \\ \sum_{i=0}^m x_i & \sum_{i=0}^m x_i^2 & \cdots & \sum_{i=0}^m x_i^{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^m x_i^n & \sum_{i=0}^m x_i^{n+1} & \cdots & \sum_{i=0}^m x_i^{2n} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^m y_i \\ \sum_{i=0}^m x_i y_i \\ \vdots \\ \sum_{i=0}^m x_i^n y_i \end{bmatrix} \quad (10)$$

式(10)或式(9)称为正规方程组。从式(10)中解出 $a_k(k=0,1,\cdots,n)$,从而可得多项式

式中, n 为该连通区域的线段数, x_1,x_2 分别为第 k 条线段两端点的 x 坐标。

$$p_n(x) = \sum_{i=0}^n a_i x^i \tag{11}$$

多项式拟合的一般方法可归纳为以下几步:

- 1) 由已知数据画出函数粗略的图形——散点图;
- 2) 计算 $\sum_{i=0}^m x_i^j$ 和 $\sum_{i=0}^m x_i^j y_i$, ($j = 0,1,\cdots,2n$);
- 3) 利用正规方程组,求出 $a_0,a_1,\cdots,a_n$;
- 4) 最后得到拟合多项式 $p_n(x) = \sum_{i=0}^n a_i x^i$;
- 5) 对不同次多项式拟合结果比较,根据拟合误差确定拟合多项式的次数 $n(n\leq 3)$ 。

2.3.3 边界连接

$a_k(k=0,1,\cdots,n)$ (即多项式系数,这里 $k < 4$) 求出来后,在坐标系(x,y)内画出曲线即为所求。

2.4 连通区域面积计算法割除残留指纹

采用线段编码标记法^[10],通过扫描基于一次分割后的图像得到各连通区域的线段表,求得各连通区域的面积,保留面积最大的指纹区域,将多个分离的残留脊线区域分割。

- 线段编码标记法的步骤如下:
- 1) 对前景像素逐行扫描得到初始线段表 $tab[k]$ 。
 - 2) 当前线段依次与上面的线段进行 8 向连通检测,并赋予相同的标记。
 - 3) 整个线段表按标记的大小重新进行排序。
- 这样图像中的各个连通区域对应线段表中连接在一起的一段,每个区域的面积可由下式得到:

$$area = \sum_{k=1}^n tab[k] \cdot (x_2 - x_1) + 1 \tag{12}$$

3 实验结论与分析

在数据库方面,由于目前并没有面向残留指纹的数据库,为了进一步验证本文所述算法的有效性,笔者在自己采集的指纹库上进行了实验测试。图像尺寸为 320×320 像素,分辨率为 500 dot/inch , 256 级灰度图像,共包含 1 800 幅各种类型的指纹图像,来自 600 个不同手指,男女各占 50%,每个手指取 3 幅图像,采集时允许任意角度的指纹旋转。在微机上用 Visual C++ 实现了本文算法,对该指纹库中 1 800 幅指纹图像进行实验测试(实验结果如图 5),从时间上看,残留脊线区域分割的平均耗时为 25 ms、文献[7]中没有给出残留脊线区域分割的平均耗时,无法比较。有明显残留指纹的图像有 1 600 幅,一般残留指纹的面积比例从 2%到 20%不等。总共有 1 450 幅能被文中方法成功分割。一些图像不能被分割的原因是正常指纹与残留指纹接触处质量差,以致在判断方向变化时出现了错误,从而导致错误分割。指纹图像分割主要是去除背景,提高细节点检测的准确率。细节点检测的准确率可用 E_1 来表示, $E_1 = (a + b)/t$,其中, a 为图像中丢失的细节点数, b 为产生的伪节点数, t 为总的细节点数,通常由手工标定。 E_1 值越小,细节点检测的准确率越高。图 5 中的 E_1 值如表 1 所示,其中 g 表示分割后检测到的真正细节点个数,从表 1 中可以看出,本文方法的 E_1 值明显比文献[7]的低。

表 1 E_1 值比较
Tab.1 E_1 comparison

图像	手工标记/ t	文献[7]方法				本文方法			
		g	a	b	E_1	g	a	b	E_1
1	28	25	3	2	0.2	27	1	1	0.07
2	18	14	5	4	0.64	16	2	2	0.25
平均	23	20	4	3	0.35	22	2	2	0.18

4 结 论

根据残留脊线与真正脊线区域的位置关系,针对残留指纹往往具有清晰的纹路而不易被分割的特

点,提出一种连续方向图法在残留指纹分割中的应用算法。该算法的改进之处体现在:1) 利用直方图滤波技术对指纹点方向图进行滤波处理,得到连续分布方向图。实践证明这种连续分布方向图过渡平滑、自然,既具有很好的连续性、渐变性和抗噪性,又

具有很高的精确度。2) 对前景区域进一步分析,根据残留指纹和正常指纹方向一般不一致的特点,运用方向信息变化较大的特性标记出残留指纹和指纹区边界点。为验证该算法的性能,将改进算法与文献[7]算法进行了对比实验,实验结果证明了本文算法的有效性。

本文算法只适合于残留区和指纹区纹理方向不一致的情况,对于残留区和指纹区纹理方向一致和残留指纹与正常指纹重叠区如何分割的问题,将是下一步工作的重点研究方向。

该算法是根据方向的变化较大来确定残留指纹和正常指纹的边界点,故对指纹的点方向图的计算很依赖,求得的方向不准确或者有偏差就会造成拟合的边界错误。但是由于指纹方向的求取算法不可能 100%准确,尤其是指纹的中心区域和三角区域附近指纹的方向变化剧烈,在对点方向图的计算精确度方面还需进一步研究。

参考文献 (References)

- [1] Mehtre B M, Murthy N N, Kapoor S. Segmentation of fingerprint images using the directional images [J]. Pattern Recognition, 1987, 20 (4) : 429-435.
- [2] Mehtre B M, Chatterjee B. Segmentation of fingerprint images-a composite method [J]. Pattern Recognition, 1989, 22 (4) : 381-385.
- [3] Bazen A M, Gerez S H. Segmentation of fingerprint images [C] // Proceedings of the 12th Annual Workshop on Circuits, Systems and Signal Processing. Veldhoven, Netherlands: Springer, 2001: 276-280.
- [4] Bazen A M, Gerez S H. Systematic methods for the computation of the directional field and singular points of fingerprints [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24 (7) : 905-919.
- [5] Zhu En, Zhang Jianmin, Yin Jianping, et al. Removing the remaining ridges in fingerprint segmentation [J]. Journal of Zhejiang University (Science A), 2006 7 (6) : 976-983. [祝恩, 张建民, 殷建平, 等. 去除残留脊线的指纹分割 [J]. 浙江大学学报 (自然科学 A), 2006 7 (6) : 976-983.]
- [6] Nie Guijun, Wu Chen. Studies on continuously distributed directional image in automated fingerprint identification system [J]. Journal of Image and Graphics, 2005, 10 (3) : 315-320. [聂桂军, 吴陈. 自动指纹识别系统中连续分布方向图的研究 [J]. 中国图象图形学报, 2005. 10 (3) : 315-320.]
- [7] Fan Guojuan, Yin Yilong. A new remaining fingerprint segmentation algorithm [J]. Journal of Image and Graphics, 2007. 12 (10) : 1893-1896. [范国娟, 尹义龙. 一种新的残留指纹图像分割方法 [J]. 中国图象图形学报, 2007, 12 (10) : 1893-1896.]
- [8] Hong L, Fei Y, Jain A. Fingerprint image enhancement: Algorithm and performance evaluation [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, 20 (8) : 777-789.
- [9] Zhan Xiaosi, Meng Xiangxu, Yin Yilong. Fingerprint segmentation algorithm based on texture analysis [J]. Computer Engineering and Applications, 2008, 44 (21) : 162-165. [詹小四, 孟祥旭, 尹义龙. 基于纹理分析的指纹图像分割算法 [J]. 计算机工程与应用, 2008, 44 (21) : 162-165.]
- [10] Lu Zongqi, Jin Dengnan. Visual C + + . NET Image Processing Program [M]. Beijing: Tsinghua University Press, 2006: 359-367. [陆宗骥, 金登南. Visual C + + . NET 图像处理编程 [M]. 北京: 清华大学出版社, 2006: 359-367.]