

# 一种改进型的 DT 网格指纹匹配法<sup>①</sup>

江 科, 郝卫东, 阚丹青, 林 川

(桂林电子科技大学 机电工程学院, 桂林 541004)

**摘 要:** DT 网格是一种分析离散点的强有力的数学算法, 因此, DT 网格已经从最初用来分析地址学问题扩展到用来分析指纹匹配问题。在传统的 DT 网格匹配法的基础上做了一系列的改进, 这种算法相对于传统的 DT 网格匹配法, 在识别率上并没有明显的突破, 但是其利用了指纹不易改变的特新, 在匹配速度上有了明显的进展。如果应用到大型的指纹数据库, 这种算法的优越性就是显而易见的了。

**关键词:** DT 网格; 指纹; 指纹匹配

## An Improved DT Grid Algorithm of Fingerprint Matching

JIANG Ke, HAO Wei-Dong, KAN Dan-Qin, LIN Chuan

(Department of Mechanical and Electrical Engineering, Guilin University of Electronic Technology, Guilin 541004, China)

**Abstract:** DT grid is a powerful algorithm use for analyze discrete points in the mathematics. So that, DT grid is use for solve the problem of the geology but now it is use for solve the problem of fingerprint too. In this paper, we make some improve based on the traditional DT grid. Although this algorithm is not better than the traditional DT grid at the accuracy of the distinguish, but it uses less time because it uses the feature of fingerprint is not easy to change. If it use in the big database, the superiority of this algorithm is very obvious.

**Key words:** DT grid; fingerprint; fingerprint matching

## 1 引言

国家在发展, 社会在进步, 身份的确认在日常生活中的作用日益明显。指纹因为其唯一性和稳定性, 已经成为确认身份的一个重要的工具, 指纹的匹配问题是指纹识别中最为关键的一个问题。因为公用指纹识别仪的数据库是一个非常庞大的数据库, 提升指纹匹配的速度也成为当今科学界的一个热门话题。

迄今为止, 许多国家的学者就指纹匹配这个问题上已经提出了各种各样的理论, 其中给予指纹特征点模式的匹配算法是最多的。罗西平等人的—种细节匹配算法引入脊线信息, 虽然引入脊线信息提高了匹配算法的有效性, 但是增大了存储要求。Jin Qi 等人定义了一个由细节点周围若干采样点组成的新特征向量, 并且结合方向场信息进行匹配。卢朝阳等人的基于 DT (Delaunay Triangulation) 网格的细节点匹配算法,

利用 DT 网格搜索匹配线对从而加快搜索速度及匹配速度。尹义龙等人提出了一种基于 DT 网格的指纹匹配算法, 它利用 DT 网格中的相似三角形计算指纹姿势纠正的参数从而进行匹配, 是一种比较有效的指纹匹配算法。本文就这个 DT 网格的算法上做了一系列的改进。

事实上, 拥有世界上最大的指纹档案库的美国中央情报局(FBI)指出: 如果两个指纹中有 12 个特征点能够一一匹配, 那么就可以断定这两个指纹来自同一个手指, 本文就以这个结论来作为判断指纹是否来自同一手指的依据。

## 2 DT网格的特性

在指纹识别中, DT 网格具有三个重要特性: 1)空外接圆性质: 是指由点集 V 生成的 DT 网格中, 每个三角

<sup>①</sup> 收稿时间:2011-12-04;收到修改稿时间:2012-01-22

形的外接圆内均不包含点集  $V$  中其他任意点; 2)最小角最大性质: 指的是由点集  $V$  所能生成的三角网格中, DT 网格中三角形的最小角的角度是最大的, 有此性质, DT 网格尽可能地避免了出线狭长, 尖锐的三角形的生成, 而这些三角形中的某个角度太小, 过于狭长和尖锐, 在判断的时候容易出现误差, 也就使得误识率增高。所以 DT 网格比起其他的三角网格例如拓扑网格, 正确率更高。(3)唯一性, 这一点是 DT 网格用于指纹识别的最重要的特性, 指的是在理想的状态下(即没有弹性形变, 没有特征点的增加或者丢失的情况, 只有角度和唯一的变化), 由点集  $V$  生成的 DT 网格是唯一的。由于这个特性, 使来自同一手指的不同指纹就算有特征点的丢失或者受到伪特征点的干扰, 也会有局部的相似性, 根据这些局部的相似性可以作为判断两指纹是否来自同一手指的先前条件。

### 3 原始的DT网格匹配法

原始的 DT 网格匹配法的关键部分由以下几个步骤组成:

1) 从两个指纹图像生成的 DT 网格中寻找相似边  $L_1$  和  $L_2$ , 如图 1 所示, 如果  $L_1$  和  $L_2$  满足条件:

$$|l_1 - l_2| < T_1 \tag{1}$$

$$\|D_{11} - D_{12} - |D_{21} - D_{22}\| < T_2 \tag{2}$$

则可以判定两天边为相似边。其中,  $l_1$  和  $l_2$  分别为  $L_1$  和  $L_2$  的边长,  $D_{11}$  和  $D_{12}$  为构成  $L_1$  的两个细节点的方向,  $D_{21}$  和  $D_{22}$  为构成  $L_2$  的两个细节点的方向,  $T_1$  和  $T_2$  为根据经验选取的阈值。

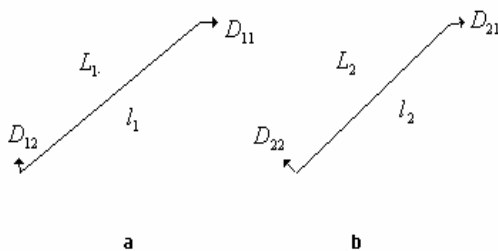


图 1 判断两条边是否相似的示意图

2) 如果  $L_1$  和  $L_2$  能够匹配, 接下来就是查看这两条边所在的三角形能否匹配, 如果都不能匹配, 则算法结束, 进入下一循环。而判断两三角形是否为相似三角形如图 2 所示。

图 2 中  $D_{11}, D_{12}, D_{21}, D_{22}$  表示细节点的方向,  $\theta_1, \theta_2, \theta_{11}, \theta_{12}, \theta_{21}, \theta_{22}$  表示内角,  $l_1, l_2, l_{11}, l_{12}, l_{21}, l_{22}$  表示边长。如果两三角形  $Tri_1$  和  $Tri_2$  满足下列条件, 则能够判断两三角形相似。

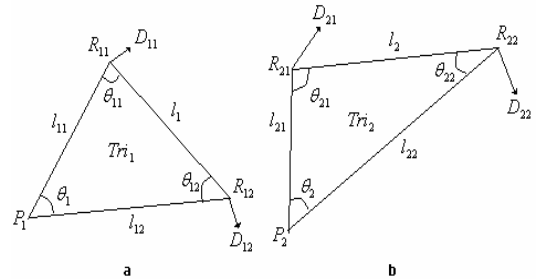


图 2 判断两三角形是否相似的示意图

$$|l_{11} - l_{21}| < T_3, |l_{12} - l_{22}| < T_3 \tag{3}$$

$$\begin{cases} |\theta_1 - \theta_2| < T_4 \\ |\theta_{11} - \theta_{21}| < T_4 \\ |\theta_{12} - \theta_{22}| < T_4 \end{cases} \tag{4}$$

3) 以已经匹配的三角形的三组对应点为参考点, 匹配其余的三角形, 直到把所有的可以匹配的三角形找到为止;

4) 根据已经匹配成功的特征点采用下列公式计算出待识别指纹相对于数据库中的原型指纹的方位位移和角度偏差;

$$\begin{bmatrix} X' \\ Y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos \Delta\theta & \sin \Delta\theta & 0 \\ \sin \Delta\theta & -\cos \Delta\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ \theta \end{bmatrix} - \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{bmatrix} \tag{5}$$

其中,  $[X \ Y \ \theta]^T$  表示的是数据库中的原型指纹中细节点的位置以及方向,  $[X' \ Y' \ \theta']^T$  表示根据原型指纹中的细节点修正后, 在待识别图像中的细节点应该在的位置, 而  $[\Delta x \ \Delta y \ \Delta\theta]^T$  就是所求的方位位移和角度偏差。然后根据得出的数据, 使用同样的公式计算出未匹配细节点的修正后的位置, 以达到进行全局矫正的目的;

5) 进行全局的匹配, 设  $M1$ =数据库中原型指纹的细节点的总点数,  $M2$ =待识别指纹细节点的总点数,  $N$ =两幅指纹图像能够匹配的细节点的总点数, 然后得到一个匹配分数  $matchscore = N/\sqrt{M1 * M2}$ , 然后根据匹配分数的值来判断是两幅指纹图像是否是同一手指的指纹。

#### 4 改进的DT网格匹配法

在原始的 DT 网格匹配法中,判断相似边就没有考虑到两条边的向量问题,在后来的三角形的匹配中也没有考虑到新的细节节点的向量问题,这就会大大的增加了匹配过程中的边和三角形的误识率,从而增加了许多不必要的计算,增加了匹配过程中所用的时间。并且,在原始的方法当中,相似边和相似三角形的匹配条件都是局限于一个根据经验得出的阈值,这在指纹识别的应用中是不正确的,因为指纹的弹性形变会根据两个细节节点之间的距离的增加而增加,但是增加的值跟两细节节点之间的距离并非成正比,所以这不应该是一个固定的值,而应该是根据某个公式得出来的值,这个公式应该根据经验反复推敲,然后无限地接近正确的结果。

本文就上述提出的问题中做了一系列的改进,并且初步推出了角度的偏差量与两细节节点之间距离的相关公式,得到了一种改进型的 DT 网格匹配法,下面我们就来具体地介绍这种方法。

##### 4.1 角度偏差的极限

这是本文提出的一个新的概念,在以后的章节中会多次用到。如图 3 所示,边 AO 与边 BO 构成的角为  $\alpha$ ,边 CD 与边 DE 构成的角为  $\beta$ ,并且 CD 的长度大于 AO。我们现在假设细节节点 O, B, D, E 在指纹提取的过程没有发生任何的弹性形变,而 A 则因为弹性形变移动到了 A' 的位置,并且 OA 与 OA' 是等长的,于是 OA' 与 OB 构成的新夹角为  $\alpha'$ ;同时 C 也因为弹性形变移动到了 C' 的位置, C'D 与 DE 构成的夹角为  $\beta'$ 。我们假设 C'D 的长度等于 CD, CC' 的长度等于 AA' 的长度,那么,根据等腰三角形的定理,我们可以推出  $\alpha - \alpha' > \beta - \beta'$ 。

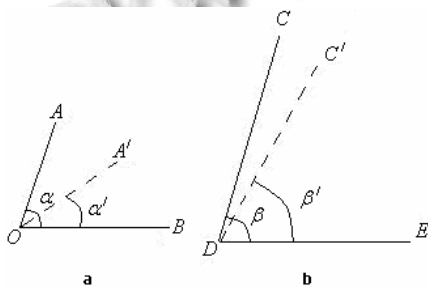


图 3 角度偏差极限的示例图

虽然弹性形变有可能随着距离的增加而增大,但

是我们这里不考虑弹性形变过大的三角形,否则很容易发生误判。所以由较长边所形成的三角形的角度偏差极限要小于由较短的边所形成的三角形的角度偏差极限,我们假设其最大位移量相等为  $\Delta l$ ,那么一个角的角度偏差极限的计算公式如下:

$$\Delta\theta_{\max} = \sum_i (2 \times a \sin(\frac{\Delta l / 2}{l_i})) \quad (i = 1, 2) \quad (6)$$

其中一条边的角度偏差极限为:

$$\Delta\theta_{i\max} = 2 \times a \sin(\frac{\Delta l / 2}{l_i}) \quad (i = 1, 2) \quad (7)$$

其中,  $l_i$  为构成角的两条边的边长。

##### 4.2 DT 网格中边的匹配

考虑到指纹在采集的过程中,手指的皮肤会产生弹性形变,并且这个弹性形变会根据距离的增加而拉长,虽然我们这里不考虑产生弹性形变过大的指纹,但是  $T_1$  和  $T_2$  不应该是一个固定的值。本文测试用的指纹库的图像时 255\*360 的大小,允许的最短位移为 3,允许的最大位移为 12,所以对于公式(1)的改进公式为:

$$T_5 = \frac{|l_2 - 15|}{15} + 3 \quad (3 \leq T_5 \leq 12) \quad (8)$$

$$|l_1 - l_2| \leq T_5 \quad (9)$$

其中,  $l_1$  和  $l_2$  分别为  $L_1$  和  $L_2$  的边长,在这里,为了提升运算速度,将  $L_2$  设定为待识别指纹图像中的一条边,于是  $T_5$  便可以经过一次性的计算就得出结果。

本文对于细节节点的匹配条件也在原始的向量匹配条件上做了一些改进,设  $D_{11}$  为数据库中的原型指纹中的某条边  $L_1$  的向量,  $D_{12}$  为待识别指纹中的某条边  $L_2$  的向量,如果构成  $L_2$  的两个细节节点为相同的细节节点,并且  $L_2$  满足条件:

$$|D_{21} - D_{22} + 180| \leq T_6 + \Delta\theta_{l2\max} \quad (10)$$

或者

$$|D_{21} - D_{22} - 180| \leq T_6 + \Delta\theta_{l2\max} \quad (11)$$

则必须跳入下一循环,因为这样的边就算匹配成功也无法与原型指纹中的模型进行绝对匹配。其中,

$T_6$  为根据经验选取的阈值, 本文的取值为 10;  $\Delta\theta_{l2max}$  为  $L_2$  的角度偏差极限。

当然, 如果构成  $L_2$  的两个细节点为不同类型的细节点, 首先必须根据细节点的类型修正细节点的方向, 然后需要满足下列条件:

$$|D_{11} - D_{l1} - D_{21} + D_{l2}| \leq T_7 + \Delta\theta_{l2max} \quad (12)$$

$$|D_{21} - D_{l1} - D_{22} - D_{l2}| \leq T_7 + \Delta\theta_{l2max} \quad (13)$$

其中  $T_7$  为根据经验选取的阈值, 本文亦取 10,  $\Delta\theta_{l2max}$  为  $L_2$  的角度偏差极限。

### 4.3 三角形的匹配

当一对边匹配通过以后, 那么这对边给大家提供的信息就是: 它们所在的三角形当中已经有两个点和一条边是可以匹配的了。根据三角形的定力, 只要与已经匹配的边的相邻的两个角近似相等, 那么就可以判断这两个三角形近似相等。当然, 在这里还不能完全判断这两个三角形是可以相互匹配的, 因为还需考虑第三个细节点的方向问题。所以, 两个三角形能够完全匹配的条件是:

$$\begin{cases} |\theta_{11} - \theta_{21}| \leq \Delta\theta_{1max} \\ \theta_{11} \times \theta_{21} > 0 \end{cases} \quad (14)$$

$$\begin{cases} |\theta_{12} - \theta_{22}| \leq \Delta\theta_{2max} \\ \theta_{12} \times \theta_{22} > 0 \end{cases} \quad (15)$$

$$|D_{13} - D_{l1} - D_{23} + D_{l2}| < T_6 + \Delta\theta_{2lmax} \quad (16)$$

其中,  $D_{l1}$  和  $D_{l2}$  必须是经过调整后的对应的方向, 不能彼此相反。这里不能考虑第三细节点的类型问题, 至少末梢点和分叉点必须互通, 因为, 在指纹提取的过程中, 因为多方面的原因, 分叉点和末梢点相互变换是很常有的事, 如图 4 中, 左边的为分叉点, 右边的为末梢点, 仅仅只有一点距离的变异。



图 4 分叉点与末梢点

考虑到细节点类型的变换问题以及在三角形匹配过程中的方向问题, 我们在细节点提取的时候规定分叉点的方向必须与其有可能变成的末梢点近似, 反之, 末梢点亦然, 如图 5 所示, 箭头所指的方向为细节点

的方向。



图 5 分叉点与末梢点的方向

当有一对三角形通过匹配条件后, 实际上已经有两对新的边通过匹配了。接下来的工作是以这两对新的边为初始条件, 继续匹配这两对边所在的另外两对三角形。按照此方法, 直至找出所有能够匹配的三角形为止, 如图 6 所示。

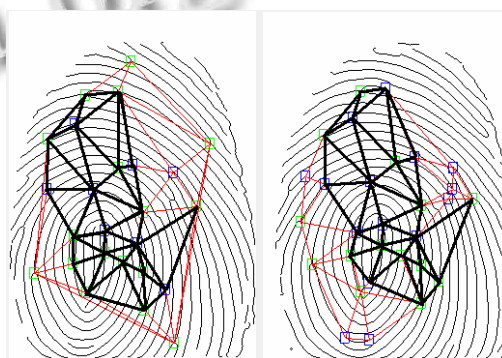


图 6 两指纹图像的 DT 网格匹配结果

这一环节中的最后的工作就是计算匹配点的数量, 第一对边的匹配点数为 2, 以后每匹配一个三角形, 匹配点的数量增 1, 但是在这里要注意匹配点的重叠。得到的匹配点的数量如果大于 12, 那么可以直接判断两指纹图像来自于同一手指了; 但是如果匹配点的数量小于 5(实际上只是相连的三对三角形匹配成功), 那么算法终止, 进入下一循环。

### 4.4 姿势的矫正以及全局的匹配

如果在 3.3 的步骤中的匹配点的数量大于等于 5 并且小于 12, 那么就要进入姿势矫正环节。首先需要求得值, 也比较容易求得为旋转角度  $\Delta\theta$ , 为了使得到的值较为准确, 我们应该用一些已经知道其向量和其在原型指纹中向量的特征信息, 利用它们的向量偏差, 求出平均值。前文已经提到, DT 网格中的各个边, 由于长度的不一样, 偏差极限也不一样, 所以不易用边的偏差量来计算, 所以只能取细节点的角度偏差来计算平均值, 但是如果有特征点变动的例如分叉点与末梢点的匹配, 这样的偏差角度也是不能用的, 因为这一类细节点的角度偏差可能会过大。所以的计算公式为:

$$\Delta\theta = \frac{\sum_i \Delta\theta_i}{n} \quad (i=1,2,3,\dots,n) \quad (17)$$

其中,  $n$  为用来计算  $\Delta\theta$  的细节点的数量,  $\Delta\theta_i$  为已经被选取的每个细节点的角度偏差。接下来便是计算  $\Delta x$  和  $\Delta y$ , 计算  $\Delta x$  和  $\Delta y$  主要利用的是细节点, 同样, 为了最终值更接近准确值, 我们应该对其做个平均化, 并且有类型变换的细节点也是不能取的。单个细节点的  $\Delta x$  和  $\Delta y$  计算公式如下:

$$\begin{bmatrix} \Delta x_i \\ \Delta y_i \end{bmatrix} = \begin{bmatrix} \cos\Delta\theta & \sin\Delta\theta \\ \sin\Delta\theta & -\cos\Delta\theta \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} - \begin{bmatrix} x'_i \\ y'_i \end{bmatrix} \quad (18)$$

( $i = 1, 2, 3, \dots, n$ )

然后在把其中的  $\Delta x_i$  和  $\Delta y_i$  之和平均化一下就可以求得  $\Delta x$  和  $\Delta y$ , 其中  $[x_i, y_i]^T$  表示的是原型指纹中细节点的位置,  $[x'_i, y'_i]^T$  为待识别图像中对应于  $[x_i, y_i]^T$  的细节点的位置。求得了  $\Delta\theta$ ,  $\Delta x$  和  $\Delta y$  这三个值之后, 就是要计算原型指纹中其他的细节点对应待识别图像中的位置以及方向, 计算公式实际上就是把公

式(17)和公式(18)整合一下, 具体形式实际上就是前文提到的公式(5)。

根据这个结果, 就可以统计待识别图像中的能够与原型指纹中的细节点匹配的细节点的个数  $N$ , 本文采取的匹配条件是如果  $N$  大于或者等于 12, 那么就判断这两个指纹来自同一手指。

## 5 实验结果

为了测试新算法的性能, 本文采用了具有 1500 个指纹图像的指纹库进行测试, 其中包含了 500 个手指, 每个手指有 3 个指纹图像, 在这 1500 个指纹图像中, 本文选取了 981 个比较不清晰的指纹图像(即 327 个手指的指纹)另外进行测试。并且, 我们运用了原始的 DT 网格算法与本文的 DT 网格算法进行对比。我们观察的数据主要有几个: 平均延时, 识别率, 拒识率, 误识率。这里的平均延时指的是把某副指纹图像与库中其他所有图像遍历比较一轮所用的时间, 识别指的是某副指纹图像与自己的同源指纹匹配得到正确的结果, 拒识所指的则相反, 误识是指两幅不同源的指纹图像被错认为来自一个指纹。

表 1 原始的 DT 网格匹配法(匹配条件, 匹配分数 > 60)

指纹图像	平均延时	识别率	拒识率	误识率
所有指纹图像	16.7 秒	93.73%(2812)	6.27%(188)	0.00000166%(15)
不清晰的图像	10.0 秒	82.47%(809)	17.53%(172)	0.0000156%(15)

表 2 原始的 DT 网格匹配法(匹配条件, 匹配数 >= 12)

指纹图像	平均延时	识别率	拒识率	误识率
所有指纹图像	16.5 秒	99.00%(2970)	1.00%(30)	0.00000356%(32)
不清晰的图像	10.0 秒	96.94%(951)	3.06%(30)	0.0000323%(31)

表 3 本文的 DT 网格匹配法(匹配条件, 匹配数 >= 12)

指纹图像	平均延时	识别率	拒识率	误识率
所有指纹图像	12.3 秒	98.90%(2967)	1.10%(33)	0.00000189%(17)
不清晰的图像	7.8 秒	96.64%(948)	3.36%(33)	0.0000167%(16)

## 6 结论

从以上三个表格可以看出, 本文的 DT 网格匹配法相对于原始的 DT 网格匹配法, 虽然拒识率略高, 但是每个指纹的平均延时大约减少了 1/4。虽然拒识率略有增高, 但是如果一个手指采集几个指纹图像作为数据库中的原型指纹便可以解决此问题。总的来说本

文的算法是可取的, 特别是用在大型的指纹库中。

### 参考文献

- 1 罗希平, 田捷. 自动指纹识别中的图像增强和细节匹配算法. 软件学报, 2002, 13(5): 946-956.
- 2 卢朝阳, 梁毅军, 寇卫东. 基于 DT 三角形网络的特征匹配方

(下转第 217 页)

表 2 特征向量矩阵

	成份	
	1	2
X1	0.867	0.423
X2	-0.804	-0.552
X3	0.901	-0.373
X4	0.871	0.167
X5	-0.303	0.857
X6	-0.488	0.739

取 2004 年至 2008 年的数据作为预测模型的训练样本,用 matlab9.0 建立 RBF 神经网络模型对北京的肺结核病的 2009 年前五个月发病人数做预测,经过不断的调整参数和反复的实验,预测的结果如表 3 所示,此外还用传统 RBF 神经网络做了对比。

表 3 预测结果

	1	2	3	4	5
实际值	0.6633	0.5899	0.7188	0.6867	0.6513
改进 RBF	0.6520	0.5466	0.7144	0.7288	0.6650
传统 RBF	0.6997	0.5145	0.7389	0.7608	0.7882

从上面的发病人数的实际值和预测值的对比结果可以看出,用训练好的 RBF 网络对北京得肺结核发病人数可以做出很好的预测,虽然个别预测值与实际值有差别,但是完全可以体现出发病的整体趋势,可以在发病趋势增大时提供预警,做好相关的预防准备。

## 5 结论

径向基神经网络属于前向神经网络类型,它以局

部响应的径向基函数代替传统的全局响应的激励函数,免了传统 BP 学习算法的收敛时间长,易陷入局部最优的缺点。径向基神经网络预测系统具有良好的性能,我们将数理统计中的主成分分析方法结合到预测模型中来,进一步提高了预测的精度。仿真结果显示,主成分-RBF 神经网络模型具有很好的预测精度,可以满足实际需求。

## 参考文献

- 1 Tian FP, Wan SH. Principal component neural network prediction model for disease prediction. Journal of Liaoning Technical University(Natural Science),2010,(5).
- 2 Zhi J, Zhang DM. Based on PCA of genetic neural network prediction of stock index. Computer Engineering and Applications,2009,45(26):210-212.
- 3 Xiang DJ. Practical Multiply Statistical Analysis. China University of Geosciences Press,2005.120-184.
- 4 Cben TP, Lin Q. New algorithm for pincipal component and minor component extraction. Journal of Fudan university (Natural Science),1997,36(2):227-230.
- 5 Moody JE, Darken C. Fast learning in networks of locally tuned processing units. Neural Computation,1989,1(2):281-294.
- 6 Andrea B, Evan B, Thomas MDA, et al. An overview and sensitivity study of a multi-mechanistic chloride transport model. Cement Concrete Res,1999,29(6):827.
- 7 Robert J, Schilling James, Carroll J. Approximation of non-linear systems with radial basis function neural networks. IEEE Trans. on Neural Networks,2001,12(1):21-28.

(上接第 109 页)

- 1 法.生物识别研究新进展(二).北京:清华大学出版社,2003. 192-196.
- 2 杨钦.限定 Delannay 三角网格剖分技术.北京:电子工业出版社,2005.
- 3 刘宁,邵晓艳,李向.DT 网格在指纹识别中的应用.河南机电

高等专科学校学报,2008,16(4):34-36.

- 4 Qi J, Wang YS. A robust fingerprint matching method. Pattern Recognition,2005,38:1665-1671.
- 5 张季.自动指纹识别算法研究与系统设计[硕士学位论文].成都:西南交通大学,2007.