

## 一种基于点模式的指纹分阶段匹配算法的研究

王峰, 王浩, 王秀友

(阜阳师范学院 计算机系, 安徽 阜阳 236041)

**摘要:**介绍了一个指纹中心点定位(Core)以及用中心点为中心构造特征向量进行初匹配,并且以此作为最佳匹配参考点来进行二次匹配的算法。本算法的特点:1. 介绍指纹中心点的准确定位。2. 以中心点作为最佳匹配参考点将匹配分为两步进行:初匹配利用了细节点间的结构关系,克服了图像的平移和旋转的影响;二次匹配引用了界限盒思想,增强指纹匹配算法对形变的适应能力。本算法把点模式的优点和基于结构的特征点之间的相对距离不变性、所跨越纹线数目的不变性、特征点类型的不变性很好结合起来。实验结果显示本算法具有较强的适应性和较高的拒识率。

**关键词:**点模式;分阶段匹配;中心点;特征向量

**中图分类号:**TP391.4

**文献标识码:**A

**文章编号:**1673-629X(2007)11-0114-03

Study of Fingerprint Grading Matching Algorithm  
Based on Point Pattern

WANG Feng, WANG Hao, WANG Xiu-you

(Dept. of Computer, Fuyang Normal College, Fuyang 236041, China)

**Abstract:** Introduce a core location and as a core to construct the feature vector first matching, and as a best matching point to make second matching. This algorithm's trait is: 1. Correctly location core; 2. Using a core as a best matching point which divide grading matching into two steps: first matching make use of minutia structure relation which overcome influence on revolving and motion; second matching excerpt bounding box idea, which enhance fingerprint matching algorithm to adapt transfiguration. This algorithm combines merit of point pattern with structure merit (feature point relative distance fixedness, spanning ridge amount fixedness, feature point type fixedness). The experiment result shows that algorithm adapts well to FAR.

**Key words:** point pattern; grading matching algorithm; core point; feature vector

## 1 指纹匹配方法综述

指纹图像的特征匹配是把新输入的指纹特征与指纹库中所存储的指纹特征相比较,计算它们的匹配程度,输出比对结果,从而判断这两枚指纹是否来自同一个指头。这里的特征通常是指脊末梢(端点)、分叉点、核心点、三角点等特征,但大部分指纹识别系统使用的都是两种典型细节点:纹线端点和纹线分叉点。纹线端点是一条纹路的终结点,而纹线分叉点是一条纹路再次分开成为两条纹路的点,如图1所示。这两种特征点在指纹图像中出现的几率最大、最稳定,易于检测,而且足以描述指纹的唯一性。



图1 指纹端点和分叉点

作为指纹识别的关键技术之一,指纹匹配算法直接决定了自动指纹识别系统最终的准确率。目前,指纹匹配算法(尤其是基于细节特征点集的指纹匹配算法)对指纹形变的适应性并不理想。点模式匹配是目前最常用的匹配方法,也是模式识别领域中一个著名的难题,它是指对于两个含有不同数量的对应点集  $P\{p_1, p_2, \dots, p_m\}$  和  $Q\{q_1, q_2, \dots, q_n\}$ , 可以通过某些变换(如平移变换、旋转变换、伸缩变换)将它们匹配起来。现在已有很多学者对此进行了研究并提出了众多方法。如松弛算法主要处理了点集间的平移,它用  $T_{ij}$  表示基于一对匹配点  $(p_i, q_j)$  的相似变换,将点模式  $P$  和  $Q$  相对于作相似变换,计算两个点模式中其他点的匹配程度,匹配程度越高,则基于匹配点对  $(p_i,$

收稿日期:2007-01-19

基金项目:安徽省教育厅资助项目(2005KJ089) 安徽省高校青年教师资助计划项目(2007jq1145)

作者简介:王峰(1973-),男,安徽太和人,硕士,讲师,主要研究方向为图像处理和模式识别。

$q_j$ ) 的  $T_{ij}$  变换的可靠度越高。该算法需要反复迭代计算  $T_{ij}$  的可靠度, 所以速度很慢。

以 Hough 变换为基础的模式匹配方法把点模式匹配转换成检测 Hough 空间中的峰值参数, 并用多种方法来降低匹配的计算复杂度。但这种方法对于较大的图像形变效果不佳, 计算量较大, 而且当特征点较少 (少于 30 个) 时, 很难在 Hough 空间里积累起足够的数据来保证一个可靠的匹配。空间里积累起足够的数据来保证一个可靠的匹配。另外, 还有一些比较有特色的方法, 如三角匹配的算法和聚类算法等, 虽然这些方法都能处理点模式之间的平移变换、旋转变换和伸缩变换, 但计算代价大, 仍然是有待解决的问题。

## 2 新匹配算法

笔者拟针对点模式匹配方法进行更深入研究, 在进一步研究分析指纹形变的数学模型之后, 提出先定位中心点, 然后再进行二次匹配, 以便最大程度上消除指纹采集过程中的形变和平移、旋转对匹配结果的影响, 该算法共分为两步:

第一步, 利用点模式思想先求出图像的中心点;

第二步,

a. 以中心点 (Core) 作为最佳匹配参考点, 将匹配分为两步进行: 初匹配利用了细节点间的结构关系, 克服了图像的平移和旋转的影响;

b. 二次匹配引用了界限盒思想, 增强指纹匹配算法对形变的适应能力, 整合多种判决条件作为识别的依据, 提高了识别率。

### 2.1 中心点的定位

中心点定义为指纹的脊曲线曲率最大的点, 在基于点模式的匹配算法中, 将中心点作为匹配的参考点具有一致性比较强的特点<sup>[1-4]</sup>。

中心点定位算法可以在方向图上进行, 如基于灰度梯度的算法, 此种算法对于拱形指纹效果不佳; 也有在细化图中跟踪得到中心点。跟踪算法比较复杂, 且对指纹质量要求高, 适用比较清晰的指纹图。如果指纹中心区噪声比较大甚至有的中心点已经丢失, 在细化图上依赖比较严格的算法往往搜不到中心点, 或搜到错误的点, 从而使这种算法的可靠性降低。若有的指纹图噪声大, 内层弧的顶点或最内层弧中的棒都缺失, 中心点就无法搜寻。

本方法分三步进行:

第一步: 中心区域的搜索。选取的中心是指纹中心的一个小区域先求指纹图像的点方向, 方法是对每个像素点求各方向上 (如图 2 所示) 相邻 8 个灰度值之和的平均值, 再求这 8 个灰度值与平均值之差的和, 最

小和所在的方向即此点所在指纹脊线的方向, 如此得到点方向图。把点方向图分为  $16 \times 16$  大小的若干块, 对每块计算直方图, 其峰值方向即为块方向, 即每块中点的主导方向, 并把这个块方向存入数组  $\text{block}[i, j]$  ( $i = 1, 2, \dots, 16; j = 1, 2, \dots, 16$ ) 中,  $i$  代表行号,  $j$  代表列号,  $\text{block}[i, j]$  即是此块的块方向编码值, 块方向编码如图 2 所示。然后在这个块方向图上按照以下原则去搜索中心区域:

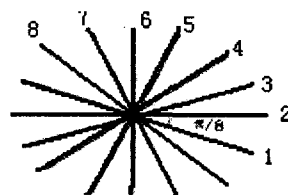


图 2 方向编码

a. 逐行检查块方向数组  $\text{block}[i, j]$ , 若  $\text{block}[i, j] = 1, 2, 3$  且  $\text{block16}[i, j+1] = 5, 6, 7$ , 则记下号;

b. 逐行检查块方向数组  $\text{block}[i, j]$ , 若  $\text{block}[i, j] = 1, 2, 3$  且  $\text{block16}[i, j+1] = 0, 1, 7$ ;  $\text{block16}[i, j+2] = 5, 6, 7$ , 则记下  $i+1, j$  号;

c. 逐行检查块方向数组  $\text{block}[i, j]$ , 若  $\text{block}[i, j] = 1, 2, 3$  且  $\text{block16}[i, j+1] = 0, 1, 7$ ;  $\text{block16}[i, j+2] = 0, 1, 7$ ;  $\text{block16}[i, j+3] = 5, 6, 7$ , 则记下  $i+1, j$  号。

$j$  做内层循环,  $i$  做外层循环, 逐行按照上述的顺序去搜索, 符合任一条则停止搜索。

第二步: 中心点初定位。搜索得到中心区域后, 再把指纹图像分割成若干个  $8 \times 8$  的小块, 求出各块的方向, 并把各块方向存入数组  $\text{block}[i, j]$  中, 然后在这个较细的块方向图上搜索中心区, 这次搜索不需对整个数组搜索, 只需在搜索得到的中心区域内搜索, 既可以避免其他区域的干扰, 又可以提高速度, 按顺序搜索, 一旦找到合适的目标则停止。

第三步: 中心点精确定位。上面求得的初步中心点, 有可能由于干扰的存在, 找到两个以上中心点, 因此必须先剔除干扰引起的中心点, 然后再进一步确定中心点的准确位置。

其步骤如下:

(1) 由上面初步确定的各中心点, 在  $8 \times 8$  的不重叠块中得到进一步搜索的范围  $\max x, \max y, \min x, \min y$ ;

(2) 在搜索范围内, 任取一块  $(i, j)$  为中心, 在其半径  $R$  的上半圆范围内计算块方向  $(k, l)$  及对应正弦值;

(3) 由于中心点定为凸脊曲率最大处, 将  $R$  上半圆划分为两个区域  $R1$  和  $R2$ , 如图 3 所示, 分别计算两

个区域内正弦和的差值;

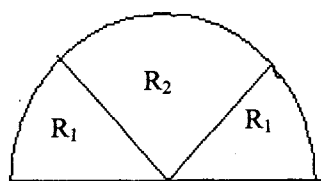


图 3 求取正弦差区域

(4)在搜索范围内,重复步骤(2)和(3),然后求出 A 最大值所对应的块,即中心点对应的块位置;

(5)在得到中心点的邻域内,将其分为  $w3 * w3$  不重叠块并求出各块的方向,重复步骤(2)、(3)、(4)。

根据需要,改变  $w3$  的大小,重复以上步骤就可以得到中心点的精确位置。

该算法将块方向信息和脊的细节特征结合起来确定中心点,相比其他定位方法,该算法对指纹图像质量要求不高,定位更准确。其中图 4 中最暗的点就代表了中心点。

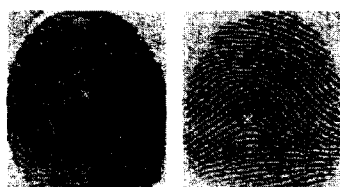


图 4 中心点定位图

## 2.2 建立特征向量

经过特征提取得到指纹图像上的所有分叉点和末梢点,每一个特征点记录了三方面信息:点的位置  $(x, y)$ ;点的类型  $s$ (分叉点,末梢点);该点所在脊线的方向角度  $\theta \in [0^\circ, 360^\circ]$ 。

在指纹图像发生平移、旋转或变形时,特征点的绝对位置信息会有很大变化。但相邻点间的距离、穿过的脊线数目和相对角度等却不会有太大的改变。因此,利用特征点间的脊线数目、方向差等结构关系建立特征向量,可以很好地解决平移及旋转问题<sup>[5,6]</sup>。

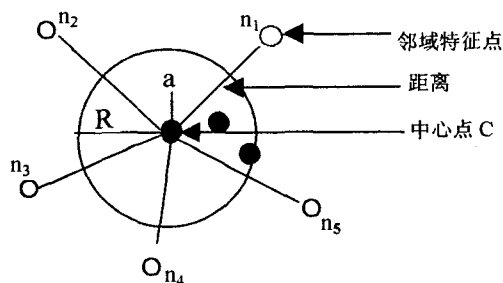


图 5 特征向量图

以中心点为中心建立一个用于匹配的邻域特征向量,结构如图 5 所示。从图中看出,取与该中心点距离大于  $R$ (这里  $R = 10$ ) 的最近 5 个点( $n1, n2, n3, n4, n5$ ) 作为其邻域特征点,距离小于等于  $R$  的点(如图

中 a 点)则不取。这 5 个邻域点和该中心点一起用来构造特征向量。每个特征向量记录的信息及其存储结构如图 5 和图 6 所示。

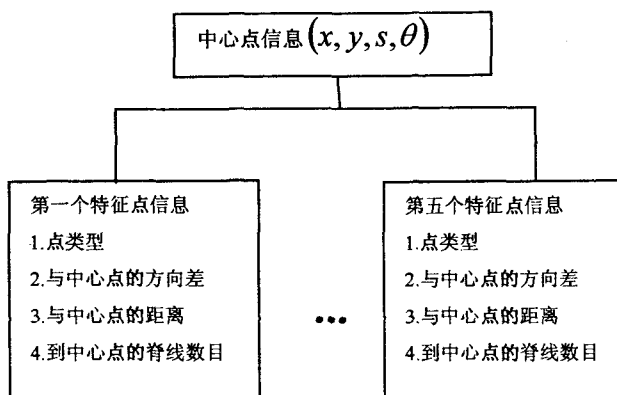


图 6 特征点数据存储类型

由于中心特征的细节类型及  $(x, y)$ 、邻域点的细节类型及  $(x, y)$  已知,各邻域点到中心点的距离以及每一邻域点与其他 4 个邻域点的最小夹角可以容易地计算出来。邻域点到中心点之间的连线所穿过的纹线数目可以根据 Bresen - ham 算法,先求出该连线所通过的各点的坐标  $(x, y)$ ,再统计值为 1 的坐标点有多少个,就可求出所穿过的纹线数目。

## 2.3 指纹图像的匹配

### 2.3.1 初匹配-特征向量匹配

假设有待识别指纹 A 和模板库中的任意指纹 B,用点集  $A = ((x_1, y_1, s_1, \theta_1), \Lambda, (x_m, y_m, s_m, \theta_m))$  表示指纹 A 上的 M 个特征点,用点集  $B = ((x'_1, y'_1, s'_1, \theta'_1), \Lambda, (x'_n, y'_n, s'_n, \theta'_n))$  表示指纹 B 上的 N 个特征点。则对指纹 A 上的每个特征点建立特征向量可得到一个 M 维的特征向量组,对指纹 B 可得到一个 N 维的特征向量组。

初匹配的过程就是首先将中心点匹配,如果待识别指纹  $A_1(x_1, y_1, s_1, \theta_1)$ (假设中心点标记为  $A_1$ ) 和模板库中的任意指纹  $B_1(x'_1, y'_1, s'_1, \theta'_1)$ (假设中心点标记为  $B_1$ ) 一致,其次把 A 中  $A_1$  与每一个特征点  $A_{i-1}(i = 1, 2, 3, \dots, M)$  构成的特征向量与 B 中  $B_1$  与每一个特征点  $B_{i-1}(i = 1, 2, 3, \dots, N)$  构成的特征向量进行一一匹配,相应的匹配分数记录在矩阵  $Score_{i \times j}$ 。匹配分数的计算方法为:若  $A_1, B_1$  的中心点类型不一致,即  $s_1 \neq s'_1$ ,则  $Score[i][j] = 0$ ;若  $A_1, B_1$  向量的中心点类型一致,即  $s_1 = s'_1$ ,且其 5 个邻域分量中有  $n$  个邻域点匹配,则  $Score[i][j] = 0 (1 \leq n \leq 5)$ 。见表 1。

如果匹配中出现测试指纹与源指纹的特征向量一对多的情况,那么找出匹配分数最高的特征向量作为匹配对。

表 1 匹配标准与匹配分数

匹配标准	匹配分数
$A(M) = B(N)$ 且 $A(m) = B(n), n = 1$	1
$A(M) = B(N)$ 且 $A(m) = B(n), n = 1, 2$	2
$A(M) = B(N)$ 且 $A(m) = B(n), n = 1, 2, 3$	3
$A(M) = B(N)$ 且 $A(m) = B(n), n = 1, 2, 3, 4$	4
$A(M) = B(N)$ 且 $A(m) = B(n), n = 1, 2, 3, 4, 5$	5
$A(M)$ :待识指纹中心点类型; $A(m)$ 待识指纹特征向量组	
$B(N)$ :模板指纹中心点类型; $B(n)$ 模板指纹特征向量组	

为减少计算量还可以引入最低匹配分数  $S(\min)$  与最高匹配分数  $S(\max)$ ,如果  $S < S(\min)$  则直接判为不匹配,不再进入第二阶段;若  $S(\min) \leq S \leq S(\max)$  则需进行二次匹配;若  $S > S(\max)$  则直接认为  $A$ 、 $B$  两点集来自同一指纹。

初匹配阶段主要完成了两个任务:

1)以指纹中心点为基准建立特征向量,加快了匹配的速度。

2)直接剔除相差较大的输入指纹图像,缩短匹配的拒绝时间,从整体上加快系统的识别速度。

### 2.3.2 二次匹配

点集  $A = (A_1, A_2, \Lambda, A_{M-1})$ 、 $B = (B_1, B_2, \Lambda, B_{N-1})$ (中心点除外),将  $A$  中的每个点与  $B$  中的每个点进行基于限界盒的一一匹配,建立  $(M-1) \times (N-1)$  维的匹配度矩阵。同一次匹配中的方法类似,为避免一点与多点相配的情况出现,采取以匹配度由高到低的原则建立  $A_{i-1}$  与  $B_{j-1}$  的一一对应。

指纹图像不可避免地存在一定范围的变形,这是一种非线性形变,一般在变形的中心区域内变化较大,然后非线性地向外扩张。因此,在判断平面中两点是否匹配时,还需要考虑一定范围内的变形,本算法引入限界盒的概念来解决这个问题。限界盒就象是放在指纹特征点上的一个盒子,限界盒的大小一般由极角和极半径来刻画,这里定义其大小为可变化的,如图 7 所示。从图中可看出,界盒的大小由当前特征点和中心点间的距离来决定,在离中心点近的地方极半径应该变小,极角变大;相反,在离中心点远的地方应该极半径变大,极角变小。因此,匹配过程中,当两个特征点落在同一个限界盒中时则认为这两个点匹配成功<sup>[7]</sup>。

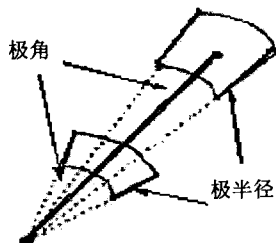


图 7 限界盒定义

### 2.3.3 匹配的判决条件

在判断两枚指纹的匹配程度时可以有不同的衡量

依据。有人认为如果两枚指纹有 12 个以上的点对互相匹配,即可认为两者来自同一个手指。而实际的输入指纹图像由于噪声和其他干扰因素的影响,很难达到这个判决门限。因此,本算法在计算匹配度矩阵时统计了以下四种信息作为判决条件:

- 1) 成功匹配的点对数;
- 2) 配对点数和相应指纹特征点总数的比值;
- 3) 各匹配点对的差异分数总和(方向差和距离差的加权和),显然,此分数越低匹配程度越高;
- 4) 差异分数总和与相应指纹特征点总数的比值。

判决时采用将以上四种条件相结合的复合判决方法,为每个条件各定义一个阈值。各条件间为并列关系,当 1~4 项信息中任意一项记录的值满足预定的阈值时,则认为两者匹配。实验结果表明:使用多判决条件的匹配方法,提高了识别率。

## 3 实验结果分析

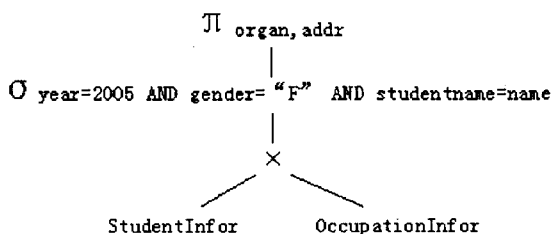
在设计的自动指纹识别系统中,图像采集器采用的是电容式晶体传感器,指纹图像的大小为  $300 \times 300$  像素,图像分辨率为 500dpi,总共从 20 个人身上采得了 500 幅指纹图像,对每个人采样 5 枚手指,从同一手指采集 5 幅图像,采集时,对指纹输入的方向没有限制。在 PIV3.06GHz 的 PC 机上,经过 10000 次匹配试验,得到以下结果:正确匹配次数 9845,正确率为 98.45%;拒识 95 次,拒识率为 0.95%,误识 60 次,误识率为 0.6%,每次识别中匹配模块用时 0.4 秒。实验证明:该算法能够解决待识图像相对于模板图像的平移、旋转等问题,不影响匹配结果。

## 4 结 论

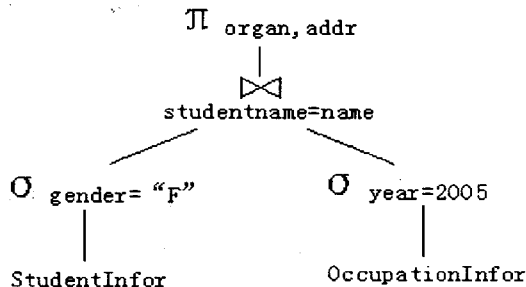
介绍了一种基于点模式的指纹特征分阶段匹配算法。该算法改进了文献[1]中使用的指纹中心点定位的方法,改进了文献[5,6]中使用的建立特征向量进行特征匹配的方法。初匹配利用了细节点间的结构关系,克服了图像的平移和旋转的影响;二次匹配引用了限界盒思想,增强指纹匹配算法对形变的适应能力,整合多种判决条件作为识别的依据,提高了识别率。本算法把点模式的优点和基于结构的特征点的相对距离不变性、所跨越纹线数目的不变性、特征点类型的不变性很好结合起来。

实验结果显示本算法具有较强的适应性和较高的拒识率。但是当指纹出现大量伪特征点和大的变形时,该算法的识别率会有所下降,所以提高抗噪和抗干扰能力,是该算法下一步改进的重点。

(下转第 124 页)



经过优化后的查询语法树为:



优化过程中,首先在可以进行选择时,倾向于连接而非乘积;其次 WHERE 子句的其他两个条件被拆分到两个选择  $\sigma$  操作中,操作被下推到各自树的相应关系上。对比发现,优化后的语法树能够节约使用的存储空间。

### 3 结束语

讨论了 Lex 和 Yacc 的基本工作原理。将它们应

用到嵌入式系统中,使得支持 SQL 语言的程序大大提高,并针对嵌入式数据库的特点,作了初步的优化工作。但要真正构造一个实用高效的 SQL 编译器,还包括 SQL 子集的详细分析、二义性的处理、语法树的进一步优化等。

#### 参考文献:

- [1] 周立功. ARM 嵌入式系统基础教程[M]. 北京:北京航空航天大学出版社, 2005:132-188.
- [2] Levine J R. Lex 与 Yacc[M]. 第 2 版. 杨作梅译. 北京:机械工业出版社, 2003:122-163.
- [3] 周立功,陈明计,陈 渝. ARM 嵌入式 Linux 系统构建与驱动开发范例[M]. 北京:北京航空航天大学出版社, 2005:71-101.
- [4] 姚泽勤,柏又青,马建峰. 利用 Lex 及 Yacc 实现嵌入式 SQL 分析器[J]. 航空计算技术, 2002(3):55-58.
- [5] 沙智华,葛研军,施志辉. 基于 Lex&Yacc 数控代码编译技术研究[J]. 组合机床与自动化加工技术, 2002(11):49-51.
- [6] 王晓东,曹庆华,王 卓. DB2 数据库查询优化策略[J]. 现代电子技术, 2006(10):92-95.
- [7] 谷震离. 关系数据库查询优化方法研究[J]. 微计算机信息, 2006(22):161-164.
- [8] 王振辉,吴广茂. SQL 查询语句优化研究[J]. 计算机应用, 2005(12):207-209.

(上接第 117 页)

#### 参考文献:

- [1] 冯俐俐,李昌禧. 指纹中心点的定位和特征匹配方法[J]. 华中科技大学学报:自然科学版, 2002(10):78-80.
- [2] 谭台哲,宁新宝,尹义龙,等. 一种基于指纹中心点的匹配算法[J]. 南京大学学报:自然科学版, 2003(4):483-490.
- [3] 朱 宁,施荣华,吴科桦. 一种新的点模式指纹匹配方法[J]. 计算机工程与应用, 2006(5):74-76.

(上接第 120 页)

系数、多项式核的指数以及核函数是值得进一步研究的问题。

#### 参考文献:

- [1] Kim K I, Park S H, Kim H J. Kernel Principal Component Analysis for Texture Classification[J]. IEEE signal processing letters, 2001, 8(2):39-41.
- [2] Haykin S. 神经网络原理[M]. 叶世伟,史忠植,译. 北京:机械工业出版社, 2004.
- [3] Vapnik V N. 统计学习理论的本质[M]. 张学工译. 北京:清

华大学出版社, 2000.

- [4] 邓乃扬,田英杰. 数据挖掘中的新方法:支持向量机[M]. 北京:科学出版社, 2004.
- [5] 张敏贵,潘 泉,张洪才,等. 基于支持向量机的人脸分类[J]. 计算机工程, 2004, 30(11):110-112.
- [6] 何国辉,甘俊英. 基于核主元分析和支持向量机的人脸识别[J]. 计算机工程与设计, 2005, 26(5):1190-1193.
- [7] 黄 勇,郑春颖,宋忠虎. 多类支持向量机算法综述[J]. 计算机技术与自动化, 2005, 24(4):61-63.
- [8] 武方方,赵银亮. 一种基于 Morlet 小波核的约简支持向量机[J]. 控制与决策, 2006, 21(8):848-852.