Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 03-034

Fingerprint Classification Using Nonlinear Discriminant Analysis

Cheonghee Park and Haesun Park

September 16, 2003

Fingerprint Classification using Nonlinear Discriminant Analysis

Cheong Hee Park Haesun Park*

Dept. of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455
chpark@cs.umn.edu hpark@cs.umn.edu

Abstract

We present a new approach for fingerprint classification based on nonlinear feature extraction. Utilizing the Discrete Fourier Transform, we construct reliable and efficient directional images to contain the representative part of local ridge orientations in fingerprints and apply kernel discriminant analysis to the constructed directional images, reducing the dimension dramatically and extracting most discriminant features. Kernel Discriminant Analysis is a nonlinear extension of Linear Discriminant Analysis (LDA) based on kernel functions. It performs LDA in the feature space transformed by a kernel-based nonlinear mapping, extracting optimal features to maximize class separability in the reduced dimensional space. We show the effectiveness of the feature extraction method in fingerprint classification. Experimental results show the proposed method demonstrates competitive performance compared with other published results.

Keywords *Discrete Fourier Transform, Fingerprint classification, Kernel discriminant analysis, Nonlinear feature extraction.*

1. Introduction

Fingerprints have long been used as a reliable biometric feature for personal identification. Fingerprint classification refers to the problem of assigning fingerprints to one of several prespecified classes. Automatic classification can be used as a preprocessing step for fingerprint matching, reducing matching time and complexity by narrowing the search space to a subset of a typically huge database [1, 2]. Fingerprints are graphical images of flow-like ridges, and they are classified as left loop (L), right loop (R), whorl (W), arch (A) and tented arch (T) according to Henry's classification scheme [3] as shown in Fig. 1.

Classification performance highly depends on the preprocessing steps where various ways to extract and represent distinguishable features among classes can be applied. The features generated after the preprocessing steps are fed into classifiers such as neural networks [4], hidden Markov model [5], probabilistic neural networks [6], support vector machines [7]. Jain et al. extracted features from Gabor-filtered images which were centered on a core point and used two-stage classifiers of K-nearest neighbor and neural networks [2]. Since the fingerprints are flow-like images, many approaches are based on directional structures explicitly or implicitly. Local orientational flows are estimated from a fingerprint image and desired features are computed from the estimated orientation field. The features can be singular points and the classification is based on the number and locations of detected singular points [8]. In [9], a directional image is partitioned into homogeneous regions and the relative relations of segmented regions are used for pattern classification. In order to overcome the possible ill-posed problem of the directional partitioning task, Cappelli et al. performed guided segmentation using directional masks as a prototype of each class [10].

In this paper, we propose a novel method for fingerprint

*This work was supported in part by the National Science Foundation grants CCR-0204109 and ACI-0305543. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation (NSF).

Figure 1. Fingerprint images. From the top left, class labels are left loop (L), right loop (R), whorl (W), arch (A), and tented arch (T).

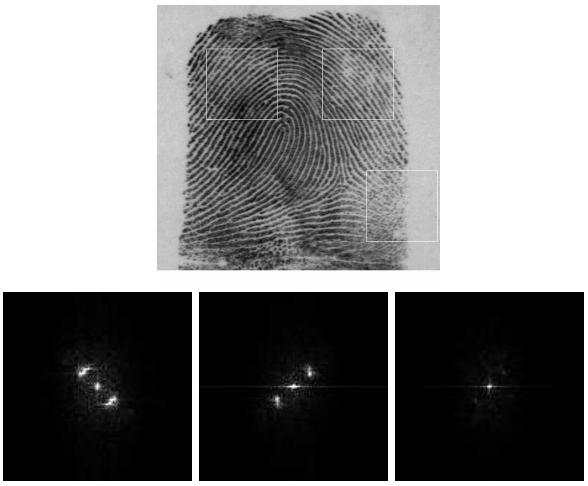

Figure 2. The fingerprint and absolute magnitude images by DFT. The values less than minimum value are displayed in black, the values greater than max/C are displayed in white, and values in between are displayed in intermediate shades of gray. The first and second pictures are DFT images of the top two squares in the fingerprint image, respectively (C=30). The last picture is the DFT image of below square in the fingerprint image (C=30).

classification using the Discrete Fourier Transform (DFT) and nonlinear discriminant analysis. In order to obtain reliable and efficient representations of fingerprint images, we utilize the Discrete Fourier Transform. The DFT is a useful computational tool that provides an efficient means for detecting directionality or periodicity in the frequency domain and removing noise by deleting high frequency coefficients. In order to estimate local ridge orientations in a fingerprint image, we apply directional filters in the frequency domain after the image has been transformed by the DFT. Directional images are then constructed in order to capture the distinguishing orientational structure among classes. By deleting high frequency coefficients in the frequency domain, our method utilizing the DFT and directional filters, can deal with noise in fingerprint images effectively, and therefore tolerate low quality fingerprint images. Kernel discriminant analysis (KDA) is a nonlinear discriminant analysis based on kernel methods. It performs Linear Discriminant Analysis after transforming the original data to a linearly separable structure using a kernel-based nonlinear mapping. Kernel discriminant analysis is capable of handling high dimensional data and extracting most discriminant features for classification automatically [12]. We apply kernel discriminant analysis to the constructed directional images extracting nonlinear discriminant features, which are used for very effective fingerprint classification.

The paper is organized as follows. In Section 2, we give a detailed explanation of how to construct directional images using the DFT. A nonlinear Discriminant Analysis based on a kernel function is presented in Section 3. Experimental results using a NIST database 4 [13] demonstrates the superior performance of our proposed method over previous methods.

2. Construction of Directional Images

Fingerprints vary in size, are positioned randomly in printed images, and the backgrounds can contain noise such as scratches, lines or blur. In the Discrete Fourier Transform (DFT), thresholding the high frequency coefficients corresponds to reducing the noise effects, while the low frequency coefficients provide a trigonometric interpolation via a finite linear combination of sines and cosines of the various frequencies of interest [11]. The DFT has been widely used in signal and image processing [14]. We apply the Fast Fourier Transform (FFT), the fast algorithm for computing the DFT, to construct a directional image.

Let I(x,y) denote the gray level at (x,y) in an $N \times N$ image. The image I in the spatial domain is transformed to the

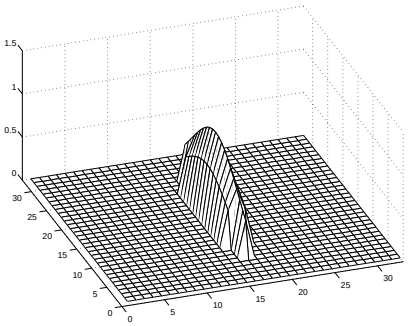

Figure 3. Directional filter D_θ of $p = 16$, $a = 1.5$, $b = 32/3$ and $\theta = 0$.

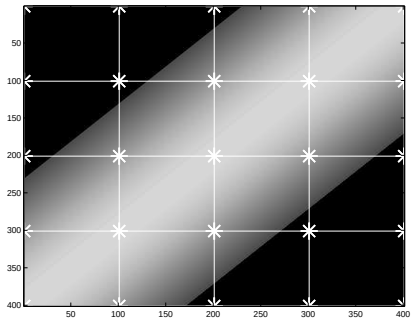

Figure 4. M_θ of $p = 2$, $a = 1.2$ and $\theta = 45^0$: The values are displayed in gray scale and pixel points of the size 5×5 are shown as $*$.

frequency domain by the 2-dimensional FFT,

$$F(k,l) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} I(m,n)\exp\{-i\frac{2\pi}{N}km\}\exp\{-i\frac{2\pi}{N}ln\}, \quad (1)$$

for $0 \le k, l \le N-1$. By translating the image F by $(\frac{N}{2}, \frac{N}{2})$ and wrapping it around at the edges, the zero frequency (DC point) is moved to the point $(\frac{N}{2}, \frac{N}{2})$ in the frequency domain and the absolute magnitude image in the frequency domain becomes symmetric around the DC point. In the rest of the paper, we will assume that the FFT has been applied and we proceed in the shifted frequency domain. Absolute magnitudes along the line passing through the DC point in the frequency domain can be viewed as responses of sine and cosine waves of the same orientation, but at various frequencies in the space domain. DFT transformed images in the frequency domain are shown in Fig. 2. The first DFT image in Fig. 2 shows a high response along the line of about 135^0 passing through the DC point, indicating the strong ridge orientation of 45^0 in the corresponding square image of the fingerprint. By using directional filters, we can find the dominant direction and its directionality.

The main steps for constructing a directional image are:

(1) Segment the fingerprint from the background by applying the FFT in a local neighborhood and computing the directionality.

(2) On a segmented fingerprint, compute the directional vectors by computing the dominant directions in a local neighborhood.

(3) Find the *core* point, to be defined later, which can be used as a landmark for unified centering of fingerprints belonging to the same class, and construct the directional image centered on the core point.

2.1 Fingerprint Segmentation

Given an input image of size 512×512, its rows and columns are divided into 16 pixels, giving 31×31 inner grid points. Apply the FFT on the 32×32 pixels centered at each inner grid point (m, n). The directional filter D_θ for orientation $\theta \in [0, 180)$ is given as follows:

$$D_\theta(i+p, j+p) = \begin{cases} \exp\{-\frac{u^2}{a^2} - \frac{v^2}{b^2}\}, & \text{if } \frac{u^2}{a^2} + \frac{v^2}{b^2} \le 1, \\ 0, & otherwise, \end{cases} \quad (2)$$

$$\text{for } -p \le i, j \le p,$$

$$\text{where } \begin{cases} v = i * cos\theta + j * sin\theta \\ u = \sqrt{i^2 + j^2 - v^2}, \end{cases} \text{ and} \quad (3)$$

$2 \times a$ and $2 \times b$ are the lengths of the short and the long axes of the ellipsoid $\frac{u^2}{a^2} + \frac{v^2}{b^2} = 1$ respectively. An example of directional filters is illustrated in Fig. 3. By using directional filters that emphasize the low frequency coefficients around the DC point and disregard the high frequency coefficients, the noise effects in fingerprint images can be reduced effectively. In our algorithm, we used $p = 16$, $a = 1.5$, $b = 32/3$ and $\theta = 0, 10, \cdots, 170$.

Directionality $\mathcal{D}_{m,n}$ and local dominant direction $\theta_{m,n} =$

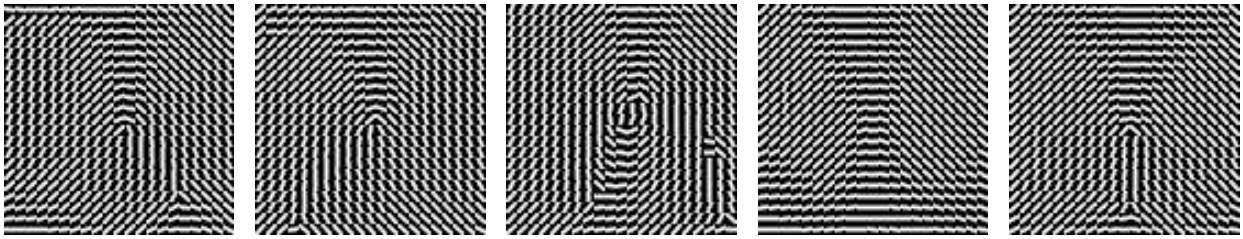

Figure 5. The directional images of size 105×105 corresponding to the fingerprints in Fig. 1. From the left, class label is L, R, W, A and T.

θ_{max} over the neighborhood are calculated as

$$\mathcal{D}_{m,n} = \sum_{\theta} \frac{f(\theta_{max}) - f(\theta)}{f(\theta_{max})} \quad (4)$$

$$\text{where} \quad f(\theta) = \sum_{i=0}^{2p} \sum_{j=0}^{2p} D_{\theta}(i,j) \times F(i,j), \quad (5)$$

$$\theta_{max} = argmax_{\theta} f(\theta). \quad (6)$$

For $\{(\mathcal{D}_{m,n}, \theta_{m,n}) \,|\, 1 \leq m \leq 31, 1 \leq n \leq 31\}$, by thresholding out the elements with low directionality $\mathcal{D}_{m,n}$ or horizontal or vertical direction $\theta_{m,n}$ and then choosing the outermost rows and columns that have the remaining elements with magnitude greater than 1, the boundaries for segmentation of foreground fingerprint from the background plane are determined. We used the mean value of $\mathcal{D}_{m,n}$, $1 \leq m, n \leq 31$ as a threshold.

2.2 Computation of Directional Vectors

The size of the segmented fingerprints can vary. In order to obtain scaling invariance, we divide the segmented fingerprint to give 31×31 inner grids and the FFT is applied to 32×32 local neighborhoods centered at each grid point. The local dominant directions are detected as θ_{max} by (5) and (6). The obtained dominant direction θ is represented as a vector $[\cos(2\theta), \sin(2\theta)]$ where $0 \leq \theta \leq 180$ to avoid the discontinuity between 0 and 180 as in [6]. Let

$$d_{m,n} = [\cos(2\theta_{m,n}), \sin(2\theta_{m,n})]$$

be directional vectors at $1 \leq m, n \leq 31$. Smoothing of the directional elements is done by averaging over a 3×3 neighborhood as follows.

$$\tilde{d}_{m,n} = [\cos(2\tilde{\theta}_{m,n}), \ \sin(2\tilde{\theta}_{m,n})], \quad (7)$$

$$\text{where} \quad [x,y] = \sum_{k=-1}^{1} \sum_{l=-1}^{1} d_{m+k,n+l}$$

$$\text{and} \quad \tilde{\theta}_{m,n} = \frac{1}{2} arctan(\frac{y}{x}).$$

2.3 Construction of Directional Image

Due to the variance in size and positions of the printed fingerprints, the computed directional array $\{\tilde{d}_{m,n} \,|\, 1 \leq m \leq 31, 1 \leq n \leq 31\}$ needs to be adjusted to increase consistency for fingerprints belonging to the same class. It can be done by detecting a point which is common in fingerprints within each class, but unique to characterize each class, and then constructing a directional image centered on the point, which is called the *core* point.

The core point is detected by measuring the consistency of directional elements over 3×3 neighborhood by

$$\| \sum_{k=-1}^{1} \sum_{l=-1}^{1} \tilde{d}_{m+k,n+l} \| \quad (8)$$

at (m,n). It measures the distance from the starting point to the finishing point after adding directional vectors within a neighborhood. The lowest value indicates inconsistency of directions over a neighborhood, as in the central point of swirling circles in the fingerprints of the class whorl (W) or a rapid turning point in orientational flows of ridges and valleys. These core points are used to locate the area which is most common within classes and discriminant between classes. Most of the information about the directional structure of a fingerprint can be found around the core point. In our experiments, 21×21 directional vectors centered on the

core point were used. From the array of size 21×21 of directional vectors, a directional image is constructed where each directional vector $\tilde{d}_{m,n}$ in (7) is drawn based on $M_{\tilde{\theta}_{m,n}}$ of size 5×5 as shown in Fig. 4, resulting in the directional map of size 105×105.

$$M_\theta(i+p, j+p) = \begin{cases} C \cdot \exp\{-\frac{u^2}{a^2}\}, & \text{if } \frac{u^2}{a^2} \le 1, \\ 0, & otherwise, \end{cases}$$
$$\text{for } -p \le i, j \le p,$$
$$\text{where} \quad \begin{cases} v = i * cos\theta + j * sin\theta \\ u = \sqrt{i^2 + j^2 - v^2}. \end{cases} \quad \text{and}$$

Here we used $p = 2$, $a = 1.2$ and $C = 10$. In Fig. 5, the constructed directional images corresponding to the fingerprints in Fig. 1 are shown, where the detected core points are located in the center.

In the next section, we present nonlinear discriminant analysis based on kernel functions and the generalized singular value decomposition (GSVD), which is called KDA/GSVD [12]. As Linear Discriminant Analysis (LDA) finds projections which maximize distances between classes and minimize scatterness within classes, KDA/GSVD performs LDA on the feature space transformed by a nonlinear mapping through kernel functions. Nonlinear mapping via a kernel function makes LDA successfully find linear projections to maximally separate classes, even if the original data is not linearly separable.

3. Kernel Discriminant Analysis based on Generalized Singular Value Decomposition

Let

$$A = [a_1, \cdots, a_n] = [A_1, \cdots, A_r] \in \mathbb{R}^{m \times n}$$

be a vector space representation of a data set A with r classes in an m-dimensional space. N_i and n_i $(1 \le i \le r)$ denote the set of column indices and the number of items in each class A_i, respectively. Then the between-class scatter matrix S_b and the within-class scatter matrix S_w are defined as

$$S_b = \sum_{i=1}^{r} n_i (c_i - c)(c_i - c)^T \quad \text{and}$$
$$S_w = \sum_{i=1}^{r} \sum_{j \in N_i} (a_j - c_i)(a_j - c_i)^T,$$

where c_i is the centroid of the i-th class and c is the global centroid.

Algorithm 1 LDA/GSVD

Given a data matrix $A \in \mathbb{R}^{m \times n}$ with r classes, it computes the $r-1$ dimensional representation y of any input vector $x \in \mathbb{R}^m$.

1. Compute $H_b \in \mathbb{R}^{m \times r}$ and $H_w \in \mathbb{R}^{m \times n}$ from A according to Eqn. (11).
2. Compute the complete orthogonal decomposition of $K = \begin{bmatrix} H_b^T \\ H_w^T \end{bmatrix} \in \mathbb{R}^{(r+n) \times m}$: $\begin{bmatrix} H_b^T \\ H_w^T \end{bmatrix} Q = P \begin{bmatrix} R & 0 \\ 0 & 0 \end{bmatrix}$.
3. Let $\gamma = \mathrm{rank}(K)$.
4. Compute W from the SVD of $P(1:r, 1:\gamma)$, which is $U^T P(1:r, 1:\gamma) W = \Sigma_b$.
5. Compute the first $r-1$ columns of $X = Q \begin{bmatrix} R^{-1}W & 0 \\ 0 & I \end{bmatrix}$, and assign them to G.
6. $y = G^T x$

Linear Discriminant Analysis (LDA) searches for a linear transformation which maximizes the between-class scatter and minimizes the within-class scatter. It is known that the optimal solution for LDA is obtained by the eigenvectors x corresponding to the $r-1$ largest eigenvalues λ of

$$S_w^{-1} S_b x = \lambda x \tag{9}$$

[15]. However, when the data dimension m is larger than the number of data n, the within-class scatter matrix S_w becomes singular and S_w^{-1} is not defined. Recently, Howland et al. have developed a method called LDA/GSVD which is a generalization of LDA based on the generalized singular value decomposition (GSVD) [16]. By using the generalized singular value decomposition (GSVD), LDA/GSVD solves a generalized eigenvalue problem

$$S_b x = \lambda S_w x \tag{10}$$

without forming the covariance matrices S_b and S_w explicitly. Using the representation of S_b and S_w,

$$S_b = H_b H_b^T, \qquad S_w = H_w H_w^T \quad \text{where} \tag{11}$$
$$H_b = [\sqrt{n_1}(c_1 - c), \cdots, \sqrt{n_r}(c_r - c)] \in \mathbb{R}^{m \times r},$$
$$H_w = [A_1 - c_1 e_1^T, \cdots, A_r - c_r e_r^T] \in \mathbb{R}^{m \times n},$$
$$e_i = [1, \cdots, 1]^T \in \mathbb{R}^{n_i \times 1},$$

the GSVD is applied to $\begin{bmatrix} H_b^T \\ H_w^T \end{bmatrix}$, diagonalizing H_b^T and H_w^T as

$$U^T H_b^T X = [\Sigma_b, 0] \quad \text{and} \quad V^T H_w^T X = [\Sigma_w, 0] \tag{12}$$

where U and V are orthogonal, X is nonsingular, and Σ_b and Σ_w are rectangular diagonal matrices. From (11) and (12), we have

$$X^T S_b X = \begin{bmatrix} \Sigma_b^T \Sigma_b & 0 \\ 0 & 0 \end{bmatrix} \text{ and } X^T S_w X = \begin{bmatrix} \Sigma_w^T \Sigma_w & 0 \\ 0 & 0 \end{bmatrix} \tag{13}$$

obtaining the leftmost $r-1$ columns of X for the LDA solution. The algorithm for LDA/GSVD is given in Algorithm 1.

Although the LDA can find the optimal transformation to preserve the cluster structure, it can not overcome the limitation due to non-linearly separable data. In order to make LDA applicable to nonlinearly separable data, a kernel-based nonlinear extension of LDA has been proposed [17, 18, 19, 20, 12]. The main idea of kernel-based methods is to map the input data to a feature space by a nonlinear mapping through a kernel function, where inner products in the feature space can be computed by a kernel function without knowing the nonlinear mapping explicitly. When the input space is transformed by a kernel-based nonlinear mapping, the dimension of the feature space becomes much larger (possibly infinite) than that of the input space, and as a result, the scatter matrices may become singular in the feature space. KDA/GSVD conducts LDA in the feature space using the generalized singular value decomposition.

Given a kernel function $\kappa(x, y)$ satisfying Mercer's condition, there exists a nonlinear mapping Φ from the original data space A to a feature space $\mathcal{F}$ such that inner products of $\Phi(x)$ and $\Phi(y)$ in the feature space $\mathcal{F}$ are computed by $\kappa(x, y)$ [21]. Let S_b^Φ and S_w^Φ be the between-class and within-class scatter matrices in $\mathcal{F}$. From the LDA/GSVD in $\mathcal{F}$, we can find an optimal transformation by the generalized eigenvectors $\psi_1, \cdots, \psi_{r-1}$ of

$$S_b^\Phi \psi = \lambda S_w^\Phi \psi. \tag{14}$$

As in (11), S_b^Φ and S_w^Φ can be expressed with H_b^Φ and H_w^Φ,

$$\begin{aligned} &S_b^\Phi = H_b^\Phi (H_b^\Phi)^T, \quad S_w^\Phi = H_w^\Phi (H_w^\Phi)^T, \quad \text{where} \\ &H_b^\Phi = [\sqrt{n_1}(c_1^\Phi - c^\Phi), \cdots, \sqrt{n_r}(c_r^\Phi - c^\Phi)] \in \mathbb{R}^{m \times r}, \\ &H_w^\Phi = [\Phi(A_1) - c_1^\Phi e_1^T, \cdots, \Phi(A_r) - c_r^\Phi e_r^T] \in \mathbb{R}^{m \times n}, \end{aligned} \tag{15}$$

and c^Φ and c_i^Φ are the global centroid and the i-th class centroid in the feature space, respectively.

Algorithm 2 KDA/GSVD

Given a data matrix $A \in \mathbb{R}^{m \times n}$ with r classes and a kernel function κ, it computes the $r-1$ dimensional representation y of any input vector $x \in \mathbb{R}^m$.

1. Compute $K_b^\Phi \in \mathbb{R}^{n \times r}$ and $K_w^\Phi \in \mathbb{R}^{n \times n}$ according to Eqns. (16).
2. Apply Algorithm 1 from Step 2 where $K = K^\Phi = \begin{bmatrix} (K_b^\Phi)^T \\ (K_w^\Phi)^T \end{bmatrix}$ to obtain $G = G^\Phi = [\tilde{\alpha}^{(1)}, \cdots, \tilde{\alpha}^{(r-1)}] \in \mathbb{R}^{n \times (r-1)}$.
3. For any input vector $x \in \mathbb{R}^{m \times 1}$, a dimension reduced representation y is computed as
$$y = (G^\Phi)^T \begin{bmatrix} \kappa(a_1, x) \\ \vdots \\ \kappa(a_n, x) \end{bmatrix}.$$

Representing $\psi = \sum_{i=1}^n \alpha_i \Phi(a_i)$, we have

$$(H_b^\Phi)^T \psi = (K_b^\Phi)^T \tilde{\alpha}, \quad (H_w^\Phi)^T \psi = (K_w^\Phi)^T \tilde{\alpha} \tag{16}$$

where $\tilde{\alpha} = [\alpha_1, \cdots, \alpha_n]^T$,

$$\begin{aligned} K_b^\Phi &= \left[\ \frac{\sqrt{n_j}}{n_j} \sum_{s \in N_j} \kappa(a_s, a_i) - \frac{\sqrt{n_j}}{n} \sum_{s=1}^n \kappa(a_s, a_i) \ \right] \\ &\qquad (1 \le i \le n, 1 \le j \le r), \\ K_w^\Phi &= \left[\ \kappa(a_i, a_j) - \frac{1}{n_s} \sum_{t \in N_s} \kappa(a_t, a_i) \ \right] \\ &\qquad (1 \le i \le n, 1 \le j \le n). \end{aligned}$$

By (16), even though the nonlinear mapping Φ and the feature space $\mathcal{F}$ are not known explicitly, LDA/GSVD can be computed in $\mathcal{F}$ through a kernel function κ. The problem is now to solve the generalized eigenvalue problem

$$K_b^\Phi (K_b^\Phi)^T \tilde{\alpha} = \lambda K_w^\Phi (K_w^\Phi)^T \tilde{\alpha}. \tag{17}$$

By applying the GSVD to $\begin{bmatrix} (K_b^\Phi)^T \\ (K_w^\Phi)^T \end{bmatrix}$, we can find the desired $r-1$ eigenvectors $\tilde{\alpha}^{(1)}, \cdots, \tilde{\alpha}^{(r-1)}$ of (17) which give the solutions $\psi_1, \cdots, \psi_{r-1}$ of (14). The algorithm for KDA/GSVD is summarized in Algorithm 2. See [12] for a detailed discussion for the KDA/GSVD method.

4. Experimental Results

The NIST database 4 is a collection of 4000 fingerprint images of size 512×512. They were classified into 5 classes,

	L	R	W	A	T
L	756	0	2	0	42
R	0	746	4	0	50
W	2	6	792	0	0
A	0	2	0	760	38
T	166	238	0	150	246

Table 1. The distribution of fingerprints in NIST database 4.

left loop (L), right loop (R), whorl (W), arch (A) and tented arch (T) by human experts. 17.5 percents of the fingerprints have multiple class labels due to a variety of ambiguities such as a scar occurring in the fingerprints, the quality of the print rolling, and the print having a ridge structure characteristic of two different classes [13]. Table 1 shows the distribution of 4000 fingerprints. Each row sums up to 800 according to the classification based on the first class label, while the natural distribution is estimated to be 0.317, 0.338, 0.279, 0.037 and 0.029 for L, R, W, A and T, respectively. Since the prediction for class T is considered most difficult and its natural distribution is relatively small compared with classes L, R and W, the uniform distribution in the NIST 4 database makes the classification more challenging. Within each row in Table 1, the distribution of fingerprints which are cross-referenced to the second class is shown. In handling the cross-referenced fingerprints, one can consider only the first class label. However, considering that the classification was difficult even by human experts, one can use multiple class labels with equal priority. The experiments used two different approaches for cross-referenced data.

Among the 4000 fingerprint images, the S0001 to S2000 fingerprints are the second rolling of the same people from whom fingerprints F0001 to F2000 were taken. For a fair comparison, we set experimental conditions the same as those in [2, 7]. The training set is formed with the first 1000 fingerprints from F labels and the first 1000 fingerprints from S labels, and the test set is composed of the remaining 2000 fingerprints. If the predicted class label for test data matches one of the multiple labels, the prediction is considered to be correct.

The first approach in our experiment is to utilize multiple class labels, resulting in the increased number of fingerprints in the training set, from 2000 to 2354. The second approach is a two-stage classifier as in [2]. The first stage is a five-class classification problem and only the first class labels are used to form the training dataset. From the first stage, the two most

rejection rate(%)	0	1.8	8.5	20.0
one-stage classifier	90.7	91.3	92.8	95.3
two-stage classifier	90.3	90.9	92.4	95.0
From (Jain et al. 1999)	-	90.0	91.2	93.5*
From (Yao et al, 2002)	-	90.0	92.2	95.6

Table 2. The prediction accuracies. Rejection rate $19.5\%^*$

	Assigned Class				
Actual Class	L	R	W	A	T
L	361	3	2	8	14
R	1	371	0	2	28
W	9	20	366	2	2
A	6	5	0	412	19
T	12	8	1	44	304

Table 3. The confusion matrix for one-stage classifier in Table 2.

probable class predictions for each test sample determine the two-class classification problem in the second-stage and we utilize multiple class labels in forming the training dataset in the second stage. However, unlike [2] we only adopt four binary classification problems, L/T, R/T, W/T and A/T at the second stage, since only the class T is overlapped with other classes significantly as shown in Table 1. This means that if T is not among two highly predicted class labels in the first stage, then the final classification is determined in the first stage without proceeding to the second stage. We call the first approach the one-stage classifier; the second approach, the two-stage classifier.

4.1. Classification Performance

Following the procedures of Section 2, we constructed directional images of 4000 fingerprints. In this experiment, we used a Gaussian Kernel for KDA/GSVD

$$\kappa(x, y) = \exp(-\frac{\|x - y\|^2}{2\sigma^2}), \sigma \in \mathbb{R} \qquad (18)$$

with 0.5 multiplied by the average of pairwise distances in the training data as the value for σ. Since there are five classes, the dimension is reduced from 105×105 to 4 by KDA/GSVD. We applied centroid-based classification in the reduced dimensional space, where each test datum is assigned to the

rejection rate(%)	0	1.8	8.5	20.0
one-stage classifier	89.7	90.5	92.9	95.6
two-stage classifier	90.6	91.2	92.8	95.5

Table 4. The prediction accuracies

	Assigned Class				
Actual Class	L	R	W	A	T
L	375	1	4	6	8
R	1	367	2	9	18
W	22	23	352	2	0
A	3	7	0	410	23
T	10	4	1	44	308

Table 5. The confusion matrix for two-stage classifier in Table 4.

class of which the centroid is the nearest to the test datum. Table 2 shows the prediction accuracy of our method as well as the results from the two-stage classifiers of [2] and support vector machines of [7]. Since Jain et al. made a rejection at the step of extracting feature vectors when the center point was detected at a corner of the image, the prediction accuracy with 0 percent rejection rate for their method is not available. In our method, the rejection criterion is based on the ratio between the distance to the nearest class centroid and the distance to the farthest class centroid. The confusion matrix by the one-stage classifier is shown in Table 3. When the test data is cross-referenced and misclassified, the first class label was considered the actual class label in the confusion matrix.

4.2. Using the Orientation Array by PCASYS

We applied nonlinear feature extraction to an array of local orientations in a fingerprint image, which was computed as in PCASYS [6]. The computational preprocess is composed of segmentation of fingerprints from the background, enhancement of the segmented fingerprint image, and the detection of local orientations of ridges and valleys and the registration of a core point, producing a two dimensional array of size 28×30 of orientation vectors $[\cos(2\theta), \sin(2\theta)]$. From the computed array, a directional image is constructed where each orientation vector $\vec{d_\theta}$ is drawn based on M_θ of the size 5×5 as shown in Fig. 4, resulting in the directional image of size 140×150. Figure 6 shows the constructed directional images. Nonlinear dimension reduction by KDA/GSVD was applied to the constructed directional images. The prediction accuracies and confusion matrix are shown in Table 4 and Table 5. This demonstrates that the nonlinear feature extraction by kernel discriminant analysis can effectively extract optimal features to maximize class separability so that high quality fingerprint classification can be obtained in the reduced dimensional space.

5. Conclusion

In this paper, a novel approach was presented for fingerprint classification based on nonlinear discriminant analysis. The directional images are constructed from fingerprint images utilizing the Discrete Fourier Transform (DFT). Applying directional filters in the frequency domain after the transformation by the DFT achieves effective low frequency filtering, reducing the noise effects in fingerprint images. The constructed directional images contain the essential directional structure which is common within each class and discriminates between classes. Kernel-based nonlinear discriminant analysis performs dramatic dimension reduction giving high quality discriminant information for classification by capturing global difference among classes. The experimental results show that even simple classification methods such as centroid-based classification can achieve high prediction accuracies over more complex methods previously published.

References

[1] H.C. Lee and R.E. Gaensslen. *Advances in Fingerprint Technology*. Elsevier, 1991.

[2] A.K. Jain, A. Prabhakar, and L. Hong. A multichannel approach to fingerprint classification. *IEEE transaction on pattern analysis and machine intelligence*, 21(4):348–359, 1999.

[3] E.R. Henry. *Classification and Uses of Finger Prints*. London: Routledge, 1900.

[4] K. Moscinska and G. Tyma. Neural network based fingerprint pattern classification. *Proc. Third Int'l Conf. Neural Network*, pages 229–232, 1993.

[5] A. Senior. A hidden markov model fingerprint classifier. *Proc. 31st Asilomar Conf. Signals, Systems, and Computers*, pages 306–310, 1997.

[6] G.T. Candela, P.J. Grother, C.I. Watson, R.A. Wilkinson, and C.L. Wilson. PCASYS-a pattern-level classification

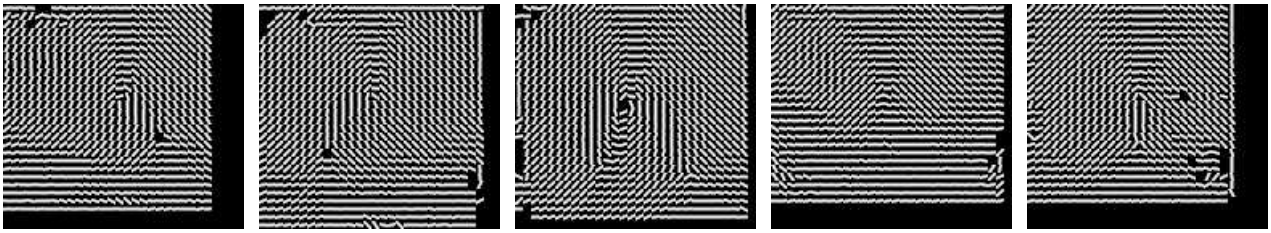

Figure 6. The directional images of size 140×150 corresponding to the fingerprints in Fig. 1. From the top left, class label is L, R, W, A and T.

automation system for fingerprints, 1995. NISTIR 5647, Aug.

[7] Y. Yao, G.L. Marcialis, M. Pontil, P. Frasconi, and F. Roli. Combining flat and structural representations for fingerprint classification with recursive neural networks and support vector machines. *Pattern Recognition*, In press, 2002.

[8] K. Karu and A.K. Jain. Fingerprint classification. *Pattern Recognition*, 29(3):389–404, 1996.

[9] D. Maio and D. Maltoni. A structural approach to fingerprint classification, 1996. Proc. 13th ICPR, Vienna, Aug.

[10] R. Cappelli, A. Lumini, and D. Maio. Fingerprint classification by directional image partitioning. *IEEE transaction on pattern analysis and machine intelligence*, 21(5):402–421, 1999.

[11] A.V. Oppenheim, A.S. Willsky, and I.T. Young. *Signals and Systems*. Prentice-Hall, 1983.

[12] C.H. Park and H. Park. Kernel discriminant analysis based on generalized singular value decomposition. Technical Reports 03-017, Department of Computer Science and Engineering, University of Minnesota, Twin Cities, 2003.

[13] C.I. Watson and C.L. Wilson. NIST special database 4. fingerprint database. National Institute of Standard and Technology(March 1992).

[14] F. Liu and R.W. Picard. Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. *IEEE transaction on pattern analysis and machine intelligence*, 18(7):722–733, 1996.

[15] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Acadamic Press, second edition, 1990.

[16] P. Howland, M. Jeon, and H. Park. Structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 25(1):165–179, 2003.

[17] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. In E.Wilson J.Larsen and S.Douglas, editors, *Neural networks for signal processing IX*, pages 41–48. IEEE, 1999.

[18] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural computation*, 12:2385–2404, 2000.

[19] V. Roth and V. Steinhage. Nonlinear discriminant analysis using kernel functions. *Advances in neural information processing systems*, 12:568–574, 2000.

[20] S.A. Billings and K.L. Lee. Nonlinear fisher discriminant analysis using a minimum squared error cost function and the orthogonal least squares algorithm. *Neural networks*, 15(2):263–270, 2002.

[21] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge, 2000.