# Kernel Fisher Discriminant Analysis with Locality Preserving for Feature Extraction and Recognition

**Di Zhang**
*School of Information Engineering, Guangdong Medical College, Song Shan Hu*
*Dongguan, Guangdong, China*
*E-mail: changnuode@163.com*

**Jiazhong He**
*Department of Physics, Shaoguan University, Da Tang Lu*
*Shaoguan, Guangdong, China*
*E-mail: hejiazhong@126.com*

**Yun Zhao**
*School of Information Engineering, Guangdong Medical College, Song Shan Hu*
*Dongguan, Guangdong, China*
*E-mail: zyun@gdmc.edu.cn*

Received 3 November 2012

Accepted 12 March 2013

## Abstract

Many previous studies have shown that class classification can be greatly improved by kernel Fisher discriminant analysis (KDA) technique. However, KDA only captures global geometrical structure and disregards local geometrical structure of the data. In this paper, we propose a new feature extraction algorithm, called locality preserving KDA (LPKDA) algorithm. LPKDA first casts KDA as a least squares problem in the kernel space and then explicitly incorporates the local geometrical structure information into the least squares problem via regularization technique. The fact that LPKDA can make full use of two kinds of discriminant information, global and local, makes it a more powerful discriminator. Experimental results on four image databases show that LPKDA outperforms other kernel-based algorithms.

*Keywords*: Kernel-based method, Fisher discriminant analysis, feature extraction, pattern classification.

## 1. Introduction

In the past two decades, appearance-based image recognition has attracted considerable interest in computer vision, machine learning, and pattern classification [1-5]. It is well known that the dimension of an image is usually very high. For example, an image with a resolution of 120×120 can be viewed as a 14400-dimensional vector. High dimensionality of feature vector has become a critical problem in practical applications. The data in the high-dimensional space is usually redundant and may degrade the performance of classifiers when the number of training samples is much smaller than the dimensionality of the image data. A common way to resolve this problem is to use feature extraction techniques. Among the enormous published feature extraction approaches, kernel-based methods, e.g., kernel principal component analysis (KPCA) and kernel Fisher discriminant analysis (KDA), have been found to be very effective in many real-world applications. KPCA was originally developed by Scholkopf et al. in 1998 [6] and KDA was introduced by Mika et al. in 1999 [7]. Subsequent research saw the development of a series of KDA algorithms (e.g.,

Co-published by Atlantis Press and Taylor & Francis

Baudat and Anouar [8], Mika et al. [9], [10], Lu et al. [11], Billings and Lee [12], Cawley and Talbot [13], Yang et al. [14], Kim et al. [15], Cortes et al. [16], and Lin et al. [17]). However, all these kernel-based methods [6-17] only capture global geometrical structure and disregard local geometrical structure of the data. If the data lies on a submanifold which reflects the inherent structure of the data space, it is difficult for these kernel-based methods to find the hidden manifold.

The theory of differential geometry shows that the manifold's intrinsic geometry can be fully determined by the local metric and the infinitesimal neighborhoods information. In view of this, some new feature extraction techniques, such as locally linear embedding (LLE) [18], Isomap [19], Laplacian eigenmap [20], graph embedding [21], and locality preserving projection (LPP) [22], have been proposed to examine the submanifold structure of the data. All such methods attempt to embed the original data into a submanifold by preserving the local geometrical structure. Different from LLE, Isomap and Laplacian eigenmap, LPP is a linear algorithm which is quite simple and easy to realize, thus has received much attention in the research community [23-30]. He et al. [23] applied LPP on the face recognition and demonstrated the effectiveness of LPP in exploring the local geometrical structure of the data. In [24], a discriminant locality preserving projection (DLPP) algorithm was proposed to improve the classification performance of LPP. Yang et al. [25] developed an unsupervised discriminant projection (UDP) technique for dimensionality reduction. An orthogonal discriminant locality projection (ODLPP) method was proposed in [26] for face recognition. However, all these methods either suffer from the small sample size problem when dealing with high dimensional data or totally neglect the class label information. To use the class label information, Cai et al. [27] proposed linear discriminant projection (LDP) method; Yang et al. [28] developed multi-manifold discriminant analysis (MMDA) algorithm; Wong and Zhao [29] proposed supervised optimal locality preserving projection (SOLPP) and normalized Laplacian-based supervised optimal locality preserving projection (NL-SOLPP) methods; Masashi Sugiyama [47] proposed Local Fisher Discriminant Analysis (LFDA) method. To address the singularity problem, Yang et al. [30] proposed a null space discriminant locality preserving projection for face recognition. The main drawback of their approach is the expensive computational cost caused by the singular value decomposition and eigenvalue decomposition in null space.

Recent work has shown that both Fisher linear discriminant analysis (LDA) and LPP can be reformulated in the regression framework based on spectral regression [28, 31, 32]. Motivated by the ideas in [7, 14, 23, 28, 31, 32], in this paper, we will develop a new feature extraction algorithm, called locality preserving KDA (LPKDA), to integrate both global and local geometrical structure information of the data. More specifically, we first cast KDA as a least squares problem in the kernel space and then use locality preserving projection as a regularization term to model local geometrical structure. The use of locality preserving projection as regularization term has been studied in [33, 34] in the context of regression and SVM. In [34], a tuning parameter was introduced to balance the tradeoff between global structure and local structure. The rest of the paper is organized as follows. In Section 2, we give a brief review of LDA and KDA. Our LPKDA algorithm is introduced in Section 3. Extensive experiments for object recognition are conducted in Section 4 to verify the efficiency of our method. Conclusion and discussion are presented in Section 5.

## 2. Outline of LDA and KDA

In this section, we first give a brief review of LDA and KDA, and then introduce an efficient two-stage method, which is crucial to the proposed LPKDA algorithm, to solve the generalized eigenvalue decomposition (GED) problem obtained by KDA.

In classification problems, given a set of *n* *d*-dimensional samples $\mathbf{x}_1$, $\mathbf{x}_2$,……$\mathbf{x}_{n,}$ belonging to *C* known classes, LDA seeks direction $\mathbf{v}$ on which the data points of different classes are far from each other while requiring data points of the same class to be close to each other [35], i.e., LDA maximizes the objective function $J(\mathbf{v})$ (also known as the Fisher's criterion ) as follows

$$J(\mathbf{v}) = \frac{\mathbf{v}^T \mathbf{S}_B \mathbf{v}}{\mathbf{v}^T \mathbf{S}_T \mathbf{v}} \tag{1}$$

$$\mathbf{S}_B = \sum_{k=1}^{C} m_k (\boldsymbol{\mu}^k - \boldsymbol{\mu})(\boldsymbol{\mu}^k - \boldsymbol{\mu})^T$$

$$\mathbf{S}_T = \sum_{i=1}^{n} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T$$

where $\boldsymbol{\mu}$ is the total sample mean vector, $\boldsymbol{\mu}^k$ is the centroid of $k$-th class, $m_k$ is the number of samples in $k$-th class, and $\mathbf{x}_i^k$ is the $i$-th sample in $k$-th class. The matrices $\mathbf{S}_B$ and $\mathbf{S}_T$ are often called the between-class scatter matrix and total scatter matrix, respectively.

Maximizing the objective function (1) is equivalent to solving the generalized eigenvalue decomposition (GED) problem

$$\mathbf{S}_B\mathbf{v} = \lambda\mathbf{S}_T\mathbf{v} \tag{2}$$

The solution of (2) can be obtained by applying an eigen-decomposition on the matrix $\mathbf{S}_T^{-1}\mathbf{S}_B$, given that $\mathbf{S}_T$ is nonsingular. Since the rank of $\mathbf{S}_B$ is bounded by $C$-1, there are at most $C$-1 eigenvectors corresponding to non-zero eigenvalues [35].

The idea of KDA is to extend LDA to a nonlinear version by using the so-called kernel trick [36]. For a given nonlinear map $\phi(\cdot)$, the $d$-dimensional input space can be mapped into the $r$-dimensional feature space, i.e.,

$$\phi : \mathrm{R}^d \to \mathrm{R}^r$$

Here, the dimension of the feature space $r$ can either be finite or infinite. Let $\boldsymbol{\mu}_\phi^k = (1/m_k)\sum_{i=1}^{m_k}\phi(\mathbf{x}_i^k)$, $\boldsymbol{\mu}_\phi = (1/n)\sum_{i=1}^{n}\phi(\mathbf{x}_i)$ and $\overline{\phi(\mathbf{x}_i)} = \phi(\mathbf{x}_i) - \boldsymbol{\mu}_\phi$ denote the centroid of the $k$-th class, the global centroid and the centered data sample in the feature space, respectively. For the new between-class scatter matrix in the feature space, following some simple algebraic steps, we see that

$$\begin{aligned}\mathbf{S}_B^\phi &= \sum_{k=1}^{C} m_k(\boldsymbol{\mu}_\phi^k - \boldsymbol{\mu}_\phi)(\boldsymbol{\mu}_\phi^k - \boldsymbol{\mu}_\phi)^T \\ &= \sum_{k=1}^{C}\frac{1}{m_k}\sum_{i=1}^{m_k}\overline{\phi(\mathbf{x}_i^k)}\,\overline{\sum_{i=1}^{m_k}\phi(\mathbf{x}_i^k)}^T \\ &= \sum_{k=1}^{C}\overline{\phi(\mathbf{X}^k)}\mathbf{W}^k\overline{\phi(\mathbf{X}^k)}^T \\ &= \overline{\phi(\mathbf{X})}\mathbf{W}\overline{\phi(\mathbf{X})}^T\end{aligned} \tag{3}$$

where $\mathbf{W}$ = diag( $\mathbf{W}^1$ , $\mathbf{W}^2$ , … $\mathbf{W}^C$ ), $\mathbf{W}^k$ is an $m_k \times m_k$ matrix with all elements equal to $1/m_k$, and $\overline{\phi(\mathbf{X}^k)} = [\overline{\phi(\mathbf{x}_1^k)},\ldots\ldots,\overline{\phi(\mathbf{x}_{m_k}^k)}]$ is the centered data matrix of the $k$ -th class in the feature space. The matrix $\mathbf{W}$ can be defined as the edge weight matrix of a graph G and its entry $\mathbf{W}_{ij}$ is the weight of edge corresponding to the vertices $i$ and $j$.

Similarly, the new total scatter matrix in the feature space can be rewritten as

$$\begin{aligned}\mathbf{S}_T^\phi &= \sum_{i=1}^{n}(\phi(\mathbf{x}_i) - \boldsymbol{\mu}_\phi)(\phi(\mathbf{x}_i) - \boldsymbol{\mu}_\phi)^T \\ &= \overline{\phi(\mathbf{X})}\,\overline{\phi(\mathbf{X})}^T\end{aligned} \tag{4}$$

By replacing $\mathbf{S}_B$ and $\mathbf{S}_T$ in (1) with $\mathbf{S}_B^\phi$ and $\mathbf{S}_T^\phi$ respectively, we obtain the corresponding objective function in the feature space as

$$J(\mathbf{v}) = \frac{\mathbf{v}^T\mathbf{S}_B^\phi\mathbf{v}}{\mathbf{v}^T\mathbf{S}_T^\phi\mathbf{v}} \tag{5}$$

However, direct calculation of $\mathbf{v}$ by solving the corresponding GED problem of (5) is difficult because the dimension of $\mathbf{v}$ is not known and furthermore it could be infinite. To resolve this problem, instead of mapping the data explicitly, an alternative way is using dot-products of the training samples to reformulate the objective function [7, 8].

Clearly, the optimal projection vector $\mathbf{v}$ is a linear combination of the centered training samples in the feature space, i.e.,

$$\mathbf{v} = \sum_{i=1}^{n}\alpha_i\,\overline{\phi(\mathbf{x}_i)} = \overline{\phi(\mathbf{X})}\boldsymbol{\alpha} \tag{6}$$

for some $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots \alpha_n]^T \in \mathrm{R}^n$. By substituting (6) into (5), following some simple algebraic steps, we see that

$$J(\mathbf{v}) = \frac{\mathbf{v}^T\mathbf{S}_B^\phi\mathbf{v}}{\mathbf{v}^T\mathbf{S}_T^\phi\mathbf{v}} = \frac{\boldsymbol{\alpha}^T\overline{\mathbf{K}}\mathbf{W}\overline{\mathbf{K}}\boldsymbol{\alpha}}{\boldsymbol{\alpha}^T\overline{\mathbf{K}}\,\overline{\mathbf{K}}\boldsymbol{\alpha}} \tag{7}$$

where $\overline{\mathbf{K}} = \overline{\phi(\mathbf{X})}^T\,\overline{\phi(\mathbf{X})}$ is a centered symmetric kernel matrix whose ($i$,$j$) element is $\overline{k(\mathbf{x}_i,\mathbf{x}_j)} = \overline{\phi(\mathbf{x}_i)}^T\,\overline{\phi(\mathbf{x}_j)}$. The optimal $\boldsymbol{\alpha}$'s can be obtained by solving the following GED problem

$$\overline{\mathbf{K}}\mathbf{W}\overline{\mathbf{K}}\boldsymbol{\alpha} = \lambda\overline{\mathbf{K}}\,\overline{\mathbf{K}}\boldsymbol{\alpha} \tag{8}$$

In [31], [37], Cai et al. developed an efficient two-stage approach to solve the generalized eigen-problem $\overline{\mathbf{X}}\mathbf{W}\overline{\mathbf{X}}^T\boldsymbol{\alpha} = \lambda\overline{\mathbf{X}}\,\overline{\mathbf{X}}^T\boldsymbol{\alpha}$, which is based on the following theorem.

**Theorem 1**. Let $\overline{\mathbf{y}}$ be the eigenvector of eigen-problem

$$\mathbf{W}\overline{\mathbf{y}} = \lambda\overline{\mathbf{y}}$$

with eigenvalue $\lambda$ . If $\overline{\mathbf{X}}^T\boldsymbol{\alpha}=\overline{\mathbf{y}}$ , then $\boldsymbol{\alpha}$ is the eigenvector of eigen-problem $\overline{\mathbf{X}}\mathbf{W}\overline{\mathbf{X}}^T\boldsymbol{\alpha}=\lambda\overline{\mathbf{X}}\overline{\mathbf{X}}^T\boldsymbol{\alpha}$ with the same eigenvalue $\lambda$ .

We could also solve the GED problem (8) efficiently by generalizing the idea presented in [31] [37] to KDA. To do so, we need the following theorem

**Theorem 2**. Let $\overline{\mathbf{y}}$ be the eigenvector of eigen-problem

$$\mathbf{W}\overline{\mathbf{y}}=\lambda\overline{\mathbf{y}} \tag{9}$$

with eigenvalue $\lambda$ . If $\overline{\mathbf{K}}\boldsymbol{\alpha}=\overline{\mathbf{y}}$ , then $\boldsymbol{\alpha}$ is the eigenvector of eigen-problem in (8) with the same eigenvalue $\lambda$ .

*Proof:* Since $\overline{\mathbf{K}}\boldsymbol{\alpha}=\overline{\mathbf{y}}$ and $\mathbf{W}\overline{\mathbf{y}}=\lambda\overline{\mathbf{y}}$ , the left side of (8) can be rewritten as

$$\overline{\mathbf{K}}\mathbf{W}\overline{\mathbf{K}}\boldsymbol{\alpha}=\overline{\mathbf{K}}\mathbf{W}\overline{\mathbf{y}}=\overline{\mathbf{K}}\lambda\overline{\mathbf{y}}=\lambda\overline{\mathbf{K}}\overline{\mathbf{y}}=\lambda\overline{\mathbf{K}}\overline{\mathbf{K}}\boldsymbol{\alpha}$$

Thus, $\boldsymbol{\alpha}$ is the eigenvector of eigen-problem in (8) with the same eigenvalue $\lambda$ .

□

Since the eigen-problem in (9) can be readily solved [31], [37], Theorem 2 shows that the KDA solution $\boldsymbol{\alpha}$ can be obtained by solving the following linear equations

$$\overline{\mathbf{K}}\boldsymbol{\alpha}=\overline{\mathbf{y}} \tag{10}$$

where $\overline{\mathbf{y}}$ is the eigenvector of **W**.

If $\overline{\mathbf{K}}$ is nonsingular, there is a unique solution $\boldsymbol{\alpha}=\overline{\mathbf{K}}^{-1}\overline{\mathbf{y}}$ for any given $\overline{\mathbf{y}}$. If $\overline{\mathbf{K}}$ is singular, however, the linear system (10) may have no solution or have infinite many solutions (the linear equation system is underdetermined). For this case, a simple and effective way is to approximate $\boldsymbol{\alpha}$ by solving the following linear equations

$$(\overline{\mathbf{K}}+\varepsilon\,\mathbf{I})\boldsymbol{\alpha}=\overline{\mathbf{y}} \tag{11}$$

where $\varepsilon\geq 0$ and **I** is the identity matrix. However, (11) is only a global approximation to (10) and local information is totally neglected. In this paper, in order to incorporate the local geometrical structure information of data sets into KDA, we use the following regularized regression problem to approximate (10)

$$\hat{\boldsymbol{\alpha}}=\arg\min_{\boldsymbol{\alpha}}\left\{\left\|\overline{\mathbf{K}}\boldsymbol{\alpha}-\overline{\mathbf{y}}\right\|^2+\varepsilon\left\|\boldsymbol{\alpha}\right\|^2\right\} \tag{12}$$

By casting KDA as a least squares problem in the kernel space, we could explicitly incorporate the local geometrical structure information into the least squares problem via regularization technique and the detail procedure is presented in the following section.

## 3. Locality Preserving KDA

KDA and its variations [23-30] only consider global geometrical structure and neglect local geometrical structure. In this section, we will develop a new KDA framework which can incorporate the local geometrical structure of data samples.

### 3.1. *Local structure modeling*

In this paper, we use LPP to model the local geometrical structure. The complete derivation and theoretical justifications of LPP can be traced back to [22]. LPP seeks to preserve local structure and intrinsic geometry of the data. The objective function of LPP is as follows

$$\frac{1}{2}\min\sum_{i,j}(y_i-y_j)^2S_{ij} \tag{13}$$

where $y_i$ is the one-dimensional projection of sample $\mathbf{x}_i$ and the matrix **S** is a similarity matrix whose element $S_{ij}$ representing the similarity between samples $\mathbf{x}_i$ and $\mathbf{x}_j$. A possible way of defining **S** is as follows

$$S_{ij}=\begin{cases}\exp(-\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2/t), & \left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2<\delta\\ 0, & \text{otherwise}\end{cases}$$

or

$$S_{ij}=\begin{cases}\exp(-\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2/t), \mathbf{x}_i\in N_k(\mathbf{x}_j)\text{ or }\mathbf{x}_j\in N_k(\mathbf{x}_i)\\ 0, \qquad\qquad \text{otherwise}\end{cases}$$

where $\mathbf{x}_i\in N_k(\mathbf{x}_j)$ implies that $\mathbf{x}_i$ is among the *k* nearest neighbors of $\mathbf{x}_j$ or vice versa [20]. The objective function incurs a heavy penalty if neighboring points are mapped far apart in the one-dimensional output space.

Since the projection of a centered sample $\overline{\phi(\mathbf{x}_i)}$ onto the vector **v** in the feature space is obtained by the inner product of **v** and the centered sample itself, we can similarly define an objective function of LPP in the feature space as follows

$$\frac{1}{2}\min\sum_{i,j}\left\|\mathbf{v}^T\overline{\phi(\mathbf{x}_i)}-\mathbf{v}^T\overline{\phi(\mathbf{x}_j)}\right\|^2S_{ij} \tag{14}$$

To gain more insight into (14), we rewrite the square of the norm in the form of matrix trace as

$$\frac{1}{2}\sum_{i,j}\left\|\mathbf{v}^T\overline{\phi(\mathbf{x}_i)}-\mathbf{v}^T\overline{\phi(\mathbf{x}_j)}\right\|^2 S_{ij}$$
$$=\frac{1}{2}\sum_{i,j}\mathrm{tr}\{(\mathbf{v}^T\overline{\phi(\mathbf{x}_i)}-\mathbf{v}^T\overline{\phi(\mathbf{x}_j)})(\mathbf{v}^T\overline{\phi(\mathbf{x}_i)}-\mathbf{v}^T\overline{\phi(\mathbf{x}_j)})^T S_{ij}\}$$
$$=\frac{1}{2}\sum_{i,j}\mathrm{tr}\{\mathbf{v}^T(\overline{\phi(\mathbf{x}_i)}-\overline{\phi(\mathbf{x}_j)})(\overline{\phi(\mathbf{x}_i)}-\overline{\phi(\mathbf{x}_j)})^T\mathbf{v}S_{ij}\}$$
$$=\frac{1}{2}\sum_{i,j}\mathrm{tr}\{\mathbf{v}^T(\overline{\phi(\mathbf{x}_i)}-\overline{\phi(\mathbf{x}_j)})S_{ij}(\overline{\phi(\mathbf{x}_i)}-\overline{\phi(\mathbf{x}_j)})^T\mathbf{v}\}\tag{15}$$

Since the operation of trace is linear and $S_{ij}$ is a scalar, Eq.(15) can be easily simplified as

$$\frac{1}{2}\sum_{i,j}\mathrm{tr}\{\mathbf{v}^T(\overline{\phi(\mathbf{x}_i)}-\overline{\phi(\mathbf{x}_j)})S_{ij}(\overline{\phi(\mathbf{x}_i)}-\overline{\phi(\mathbf{x}_j)})^T\mathbf{v}\}$$
$$=\frac{1}{2}\mathrm{tr}\left\{\mathbf{v}^T\left(\sum_{i,j}(\overline{\phi(\mathbf{x}_i)}-\overline{\phi(\mathbf{x}_j)})S_{ij}(\overline{\phi(\mathbf{x}_i)}-\overline{\phi(\mathbf{x}_j)})^T\right)\mathbf{v}\right\}$$
$$=\frac{1}{2}\mathrm{tr}\left\{\mathbf{v}^T\left(2\sum_{i,j}\overline{\phi(\mathbf{x}_i)}S_{ij}\overline{\phi(\mathbf{x}_i)}^T-2\sum_{i,j}\overline{\phi(\mathbf{x}_i)}S_{ij}\overline{\phi(\mathbf{x}_j)}^T\right)\mathbf{v}\right\}$$
$$=\frac{1}{2}\mathrm{tr}\left\{\mathbf{v}^T\left(2\overline{\phi(\mathbf{X})}\mathbf{D}\overline{\phi(\mathbf{X})}^T-2\overline{\phi(\mathbf{X})}\mathbf{S}\overline{\phi(\mathbf{X})}^T\right)\mathbf{v}\right\}$$
$$=\mathrm{tr}\left\{\mathbf{v}^T\overline{\phi(\mathbf{X})}\mathbf{L}\overline{\phi(\mathbf{X})}^T\mathbf{v}\right\}\tag{16}$$

where $\mathbf{D}=\mathrm{diag}(D_{11},\cdots,D_{nn})$, $D_{ii}=\sum_{j=1}^{n}S_{ij}(i=1,\cdots,n)$ and **L**=**D-S** is called the Laplacian matrix.

Substituting (7) into (16), we have the final form of the objective function of LPP in the kernel space

$$\begin{aligned}&\frac{1}{2}\min\sum_{i,j}\left\|\mathbf{v}^T\overline{\phi(\mathbf{x}_i)}-\mathbf{v}^T\overline{\phi(\mathbf{x}_j)}\right\|^2 S_{ij}\\&=\min\ \mathrm{tr}\left\{\boldsymbol{\alpha}^T\overline{\mathbf{K}}\mathbf{L}\overline{\mathbf{K}}\boldsymbol{\alpha}\right\}\end{aligned}\tag{17}$$

### 3.2. *Locality preserving KDA algorithm*

For the eigen-problem $\mathbf{W}\overline{\mathbf{y}}=\lambda\overline{\mathbf{y}}$, given an eigenvector $\overline{\mathbf{y}}$ with eigenvalue $\lambda$, our locality preserving KDA algorithm calculates an optimal projection vector **v** whose expansion coefficients, $\boldsymbol{\alpha}=[\alpha_1,\alpha_2,\ldots\alpha_n]^T\in\mathrm{R}^n$, are obtained from the following optimization problem:

$$\hat{\boldsymbol{\alpha}}=\arg\min_{\boldsymbol{\alpha}}\left\{\left\|\overline{\mathbf{K}}\boldsymbol{\alpha}-\overline{\mathbf{y}}\right\|^2+(1-\varepsilon)\mathrm{tr}\left\{\boldsymbol{\alpha}^T\overline{\mathbf{K}}\mathbf{L}\overline{\mathbf{K}}\boldsymbol{\alpha}\right\}+\varepsilon\left\|\boldsymbol{\alpha}\right\|^2\right\}\tag{18}$$

where $\varepsilon\in(0,1)$ is a tuning parameter that controls the tradeoff between global and local geometrical structures.

Since **W** is a block-diagonal matrix with *C* blocks, and the rank of each block is 1，there are exactly *C* eigenvectors, $\overline{\mathbf{y}}_1,\overline{\mathbf{y}}_2,\cdots\overline{\mathbf{y}}_C$, for the eigen-problem $\mathbf{W}\overline{\mathbf{y}}=\lambda\overline{\mathbf{y}}$. As a result, there are *C* optimization problems like (18) needed to be solved. For simplicity, all these optimization problems can be written in a single matrix form as

$$\begin{aligned}\hat{\mathbf{A}}=\arg\min_{\mathbf{A}}\{&\left\|\overline{\mathbf{K}}\mathbf{A}-\overline{\mathbf{Y}}\right\|^2\\&+(1-\varepsilon)\mathrm{tr}\left\{\mathbf{A}^T\overline{\mathbf{K}}\mathbf{L}\overline{\mathbf{K}}\mathbf{A}\right\}+\varepsilon\left\|\mathbf{A}\right\|_F^2\}\end{aligned}\tag{19}$$

where $\mathbf{A}=[\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\ldots\boldsymbol{\alpha}_C]$, $\overline{\mathbf{Y}}=[\overline{\mathbf{y}}_1,\overline{\mathbf{y}}_2,\cdots\overline{\mathbf{y}}_C]$,and $\|\cdot\|_F$ is the Frobenius norm of a matrix.

By differentiating the right part of Eq.(19) with respect to **A**, setting the derivative equal to zero, after some manipulation, we get

$$\overline{\mathbf{K}}^2\mathbf{A}+(1-\varepsilon)\overline{\mathbf{K}}\mathbf{L}\overline{\mathbf{K}}\mathbf{A}+\varepsilon\,\mathbf{A}=\overline{\mathbf{K}}\overline{\mathbf{Y}}\tag{20}$$

To solve (20), we need the following theorem

**Theorem 3**. Matrix $\overline{\mathbf{K}}^2+(1-\varepsilon)\overline{\mathbf{K}}\mathbf{L}\overline{\mathbf{K}}+\varepsilon\,\mathbf{I}$ is nonsingular.

*Proof:* Let $\mathbf{F}=\overline{\mathbf{K}}^2+(1-\varepsilon)\overline{\mathbf{K}}\mathbf{L}\overline{\mathbf{K}}$. By the definition of Laplacian matrix **L**, it is easy to verify that **L** is a symmetric positive semi-definite matrix [38]. With Schur decomposition, we get

$$\mathbf{L}=\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^T\tag{21}$$

where $\boldsymbol{\Lambda}=\mathrm{diag}(\lambda_1,\lambda_2,\cdots\lambda_n)$ is a diagonal matrix. Let $\mathbf{P}=\mathbf{Q}\boldsymbol{\Lambda}^{1/2}$, we have $\mathbf{L}=\mathbf{P}\mathbf{P}^T$. Thus **F** can be rewritten as

$$\mathbf{F}=\overline{\mathbf{K}}^2+(1-\varepsilon)\overline{\mathbf{K}}\mathbf{P}\mathbf{P}^T\overline{\mathbf{K}}=\overline{\mathbf{K}}^2+(1-\varepsilon)\overline{\mathbf{K}}\mathbf{P}\left(\overline{\mathbf{K}}\mathbf{P}\right)^T\tag{22}$$

From (22) it is clear that **F** is symmetric positive definite. By Cholesky decomposition, **F** can further be simplified as

$$\mathbf{F}=\mathbf{G}\mathbf{G}^T\tag{23}$$

Let $\mathbf{G}=\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ be the singular value decomposition of **G**, we have

$$\mathbf{F}+\varepsilon\,\mathbf{I}=\mathbf{G}\mathbf{G}^T+\varepsilon\,\mathbf{I}=\mathbf{U}\boldsymbol{\Sigma}^2\mathbf{U}^T+\varepsilon\,\mathbf{I}=\mathbf{U}(\boldsymbol{\Sigma}^2+\varepsilon\,\mathbf{I})\mathbf{U}^T\tag{24}$$

Thus

$$\left|\overline{\mathbf{K}}^2 + (1-\varepsilon)\overline{\mathbf{K}}\mathbf{L}\overline{\mathbf{K}} + \varepsilon\,\mathbf{I}\right|$$

$$= \left|\mathbf{U}(\mathbf{\Sigma}^2 + \varepsilon\,\mathbf{I})\mathbf{U}^T\right| = \left|\mathbf{\Sigma}^2 + \varepsilon\,\mathbf{I}\right|$$

which is nonsingular because $\varepsilon > 0$.

□

With Theorem 3, the optimal solution can be computed as

$$\hat{\mathbf{A}} = \left(\overline{\mathbf{K}}^2 + (1-\varepsilon)\overline{\mathbf{K}}\mathbf{L}\overline{\mathbf{K}} + \varepsilon\,\mathbf{I}\right)^{-1}\overline{\mathbf{K}}\overline{\mathbf{Y}} \tag{25}$$

**Algorithm: LPKDA**

Summarizing the previous subsections, the LPKDA algorithm is as follows

- Training:
  1. Generate a centered kernel matrix $\overline{\mathbf{K}} = \overline{\phi(\mathbf{X})}^T\,\overline{\phi(\mathbf{X})}$ from the training samples.
  2. Solve the eigen-problem (9) to get $\overline{\mathbf{Y}}$.
  3. Use (25) to compute **A**.
  4. Obtain a nonlinear feature matrix **Z** of the training data by $\mathbf{Z} = \mathbf{A}^T\overline{\mathbf{K}}$.
- Test:
  1. For a test sample **x**, generate a centered kernel vector $\overline{\mathbf{k}(\mathbf{x})} = \left[\overline{k(\mathbf{x},\mathbf{x}_1)}, \overline{k(\mathbf{x},\mathbf{x}_2)}, \ldots, \overline{k(\mathbf{x},\mathbf{x}_n)}\right]^T$, where $\overline{k(\mathbf{x},\mathbf{x}_i)} = \overline{\varphi(\mathbf{x})}^T\,\overline{\varphi(\mathbf{x}_i)}$.
  2. Obtain a nonlinear feature vector of the test sample by $\mathbf{z} = \mathbf{A}^T\overline{\mathbf{k}(\mathbf{x})}$

In LPKDA, the kernel function $k(\cdot,\cdot)$ plays an important role. The essential property of the kernel function is that it should be decomposed into an inner product of a mapping $\varphi(\cdot)$ to itself, i.e., $k(\mathbf{x}_i,\mathbf{x}_j) = \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$. However, it is obviously that not all the functions meet this property. To be a proper kernel function, a function should meet the so-called *Mercer's* condition [36]. The two most popular kernels are the polynomial kernel $k(\mathbf{x}_i,\mathbf{x}_j) = (\mathbf{x}_i{}^T\mathbf{x}_j + c)^d$ and the Gaussian RBF kernel $k(\mathbf{x}_i,\mathbf{x}_j) = \exp(-\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2/\sigma)$ in which *c*, *d*, and $\sigma$ are the kernel parameters.

In the training of the proposed algorithm, the most time consuming part is Step 3 where the matrix inverse problem should be solved. Because the matrices $\overline{\mathbf{K}}$ and $\mathbf{L}$ in (25) are $\mathbf{R}^{n\times n}$, the computational complexity of Step 3 is normally $O(n^3)$. Nevertheless, it is unnecessary to compute the matrix inverse involved in (25) directly. The detailed efficient procedure is discussed as follows.

Since $\mathbf{A} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots \boldsymbol{\alpha}_C]$ and $\overline{\mathbf{Y}} = [\overline{\mathbf{y}}_1, \overline{\mathbf{y}}_2, \cdots \overline{\mathbf{y}}_C]$, let $\mathbf{H} = \overline{\mathbf{K}}^2 + (1-\varepsilon)\overline{\mathbf{K}}\mathbf{L}\overline{\mathbf{K}} + \varepsilon\,\mathbf{I}$ and $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \ldots \mathbf{p}_C] = [\overline{\mathbf{X}}\overline{\mathbf{y}}_1, \overline{\mathbf{X}}\overline{\mathbf{y}}_2, \ldots \overline{\mathbf{X}}\overline{\mathbf{y}}_C]$, (25) can be decomposed into the following *C* linear equations:

$$\mathbf{H}\boldsymbol{\alpha}_i = \mathbf{p}_i, i = 1,2,\ldots C \tag{26}$$

There are many efficient iterative algorithms have been proposed to solve Eq. (26). In this paper, we use LSQR algorithm, an iterative algorithm designed to solve large scale sparse linear equations and lest squares problems [39]. In each iteration, LSQR needs to compute two matrix-vector products [40]. The computational complexity of LSQR for solving (26) is normally $O(n^2+n)$. If the sample number is large and parallel computation is applicable, using LSQR algorithm will be more efficient than performing matrix inverse directly.

## 4. Experimental results

In this section, two experiments are designed to evaluate the performance of the proposed algorithm. The first experiment is on face recognition and the second is on artificial object recognition. Face recognition is performed on three face databases (Yale, ORL, and PIE) and artificial object recognition is performed on COIL20 image database [41]. In all the experiments, we use Euclidean metric and nearest neighbor classifier for classification. In order to get a fair result, for all experiments, we adopt a two-stage scheme: 1) perform model selection, i.e., to determine the proper parameters for all the involved algorithms; and 2) reevaluate all the methods with the parameters got in the stage of model selection. Both the two stages are carried on the same data sets but under different partitions. The implementation environment is the personal computer with Intel(R) Core(TM) 2 Duo CPU P8700 @ 2.53GHz, 4 GB memory.

### 4.1. *Experiment on face recognition*

The Yale face database [42] contains 165 grayscale images of 15 individuals. There are 11 images per subject, one per different facial expressions or lighting conditions. The images demonstrate variations in lighting conditions (left-light, center-light, right-light), facial expressions (normal, happy, sad, sleep, surprised, and wink), and with/without glasses.

The ORL face database [43] has a total number of 400 images of 40 people. There are ten different images per subject. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All the images were taken with a tolerance for some tilting and rotation.

The CMU PIE database [44] contains 68 subjects with 41,368 face images as a whole. The face images were captured by 13 synchronized cameras and 21 flashes, under varying pose, illumination and expression. We choose the five near frontal poses (C05, C07, C09, C27, C29) and use all the 11,544 images under different illuminations and expressions.

Table 1. Random partition on three databases for the stage of model selection and performance evaluation.

| Database | Classes ( $C$ ) | Different numbers for training ( $n$ per subject ) | |
|---|---|---|---|
| | | Model selection | Performance evaluation |
| Yale | 40 | 5 | 2/3/5/6 |
| ORL | 15 | 5 | 2/3/5/6 |
| PIE | 68 | 60 | 30/60/90/120 |

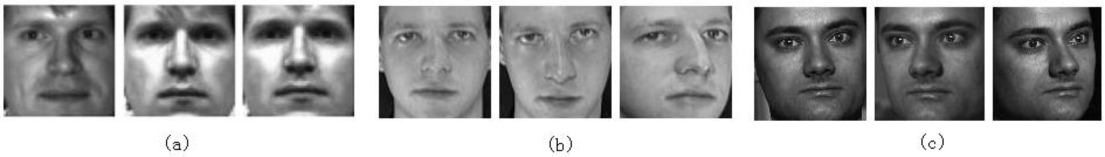


Fig.1. Samples from three face databases with, (a) Yale, (b) ORL, (c) PIE.

Table 2. Optimal parameters of each method.

| Method | KPCA | KDA | CKFD | LPP | KLFDA | LPKDA |
|---|---|---|---|---|---|---|
| parameters | [62, 8] | [27, 6] | [21, 5, 0.9 ] | [179 ] | [162, 7] | [33, 7, 0.8] |

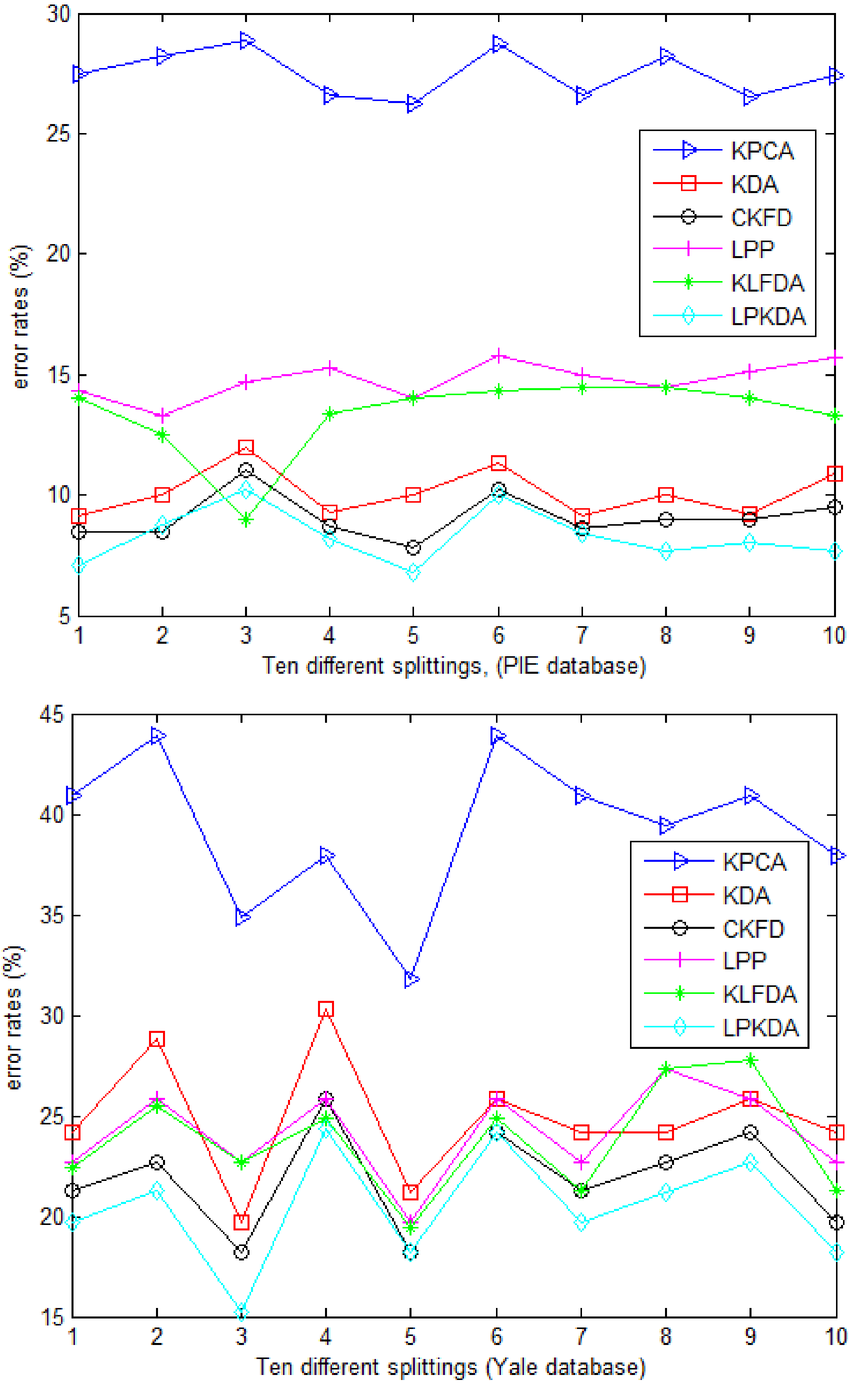
(Note that the parameter set is arranged as [subspace dimension, kernel width, coefficients].)

In our experiments, all the images are manually aligned, cropped and resized to have a resolution of $32\times 32$ pixels. Fig.1 shows some examples where three sample images of one subject are randomly chosen from each database. For each database, we randomly partition the images into a training set (*n* images per subject for training) and a test set (the remaining images are used for testing). The detailed description of partition for the stage of model selection and performance evaluation is listed in Table 1. The partition procedure is repeated 20 times and we obtain 20 different training and testing sample sets. The first 10 are used for the stage of model selection and the others for the stage of performance evaluation.

In this paper, the Gaussian RBF kernel $k(\mathbf{x},\mathbf{y})=\exp(-\|\mathbf{x}-\mathbf{y}\|^2/\sigma)$ is used. Six methods, namely, KPCA [45], KDA [45], complete kernel Fisher discriminant analysis (CKFD) [14], LPP [22], kernel local Fisher discriminant analysis (KLFDA), and the proposed LPKDA are tested and compared.

In the stage of model selection, our goal is to determine proper kernel parameters (i.e., the width $\sigma$ of the Gaussian RBF kernel), the dimension of the projection subspace for each method, the fusion coefficient that determines the weight ratio between regular and irregular discriminant information for CKFD [14], and the tuning parameter $\varepsilon$ that controls the tradeoff between global and local geometrical structure information in our proposed algorithm. Since it is very difficult to determine these parameters at the same time, a stepwise selection strategy is more feasible and thus is adopted here [11, 14]. Specifically, we fix the subspace dimension and the tuning parameter $\varepsilon$ or the fusion coefficient (only for LPKDA or CKFD) in advance and try to find the optimal kernel parameter for the Gaussian RBF kernel function. To get the proper kernel parameter, we use the global-to-local search strategy [46]. Then, based on the chosen kernel parameter, we can choose the optimal subspace dimension for each method. Finally, the tuning parameter $\varepsilon$ or the fusion coefficient is determined with respect to the other chosen parameters. After model selection, we determine all parameters for each method. Table 2 lists the parameters of each method for PIE database. With these parameters, all methods are reevaluated using another 10 sets of training and testing samples.

The error rates of random 10 different splits on three face databases with KPCA, KDA, CKFD, LPP, KLFDA and the proposed LPKDA are presented in Fig.2. The training size used in Fig.2 is 5, 5, and 30 per subject for Yale, ORL, and PIE, respectively. From Fig.2, we can see two obvious conclusions as follows:



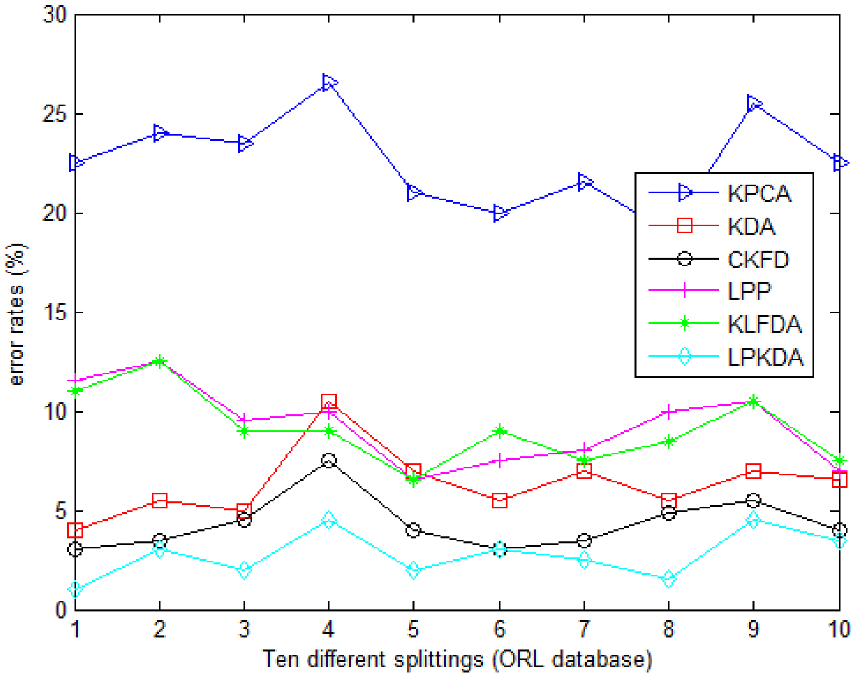


Fig.2. Comparison of KPCA, KDA, CKFD, LPP, KLFDA, and LPKDA in error rates on three face databases.

1) KPCA has the lowest performance among all the tested methods. This is because unlike other methods, KPCA yields projection directions which have minimal reconstruction error by describing as much variance of the data as possible, thus the yielded directions are meant for reconstruction, not for classification.
2) the result of KLFDA is almost the same as that of LPP, and they are slightly better than KDA on Yale database, while KDA outperforms LPP and KLFDA on ORL and PIE database. This implies that the relative importance of local and global structures in object recognition depends on specific data sets. For example, the local structure may contain less effective discriminative information in ORL and PIE database than in Yale database. However, for all the three data sets, our proposed LPKDA algorithm outperforms LPP, KDA, KLFDA, and CKFD. This demonstrates that local and global structures are complementary to each other, and better results can be achieved by properly fusing both of the local and global geometrical structure information.

We then provide detailed performance comparison of KPCA, KDA, CKFD, LPP, KLFDA, and LPKDA in Tables 3-5, where the mean error rates and standard deviations of the 10 different partitions on each data set with different training numbers are reported. It can be concluded that the proposed LPKDA achieves the best performance. From Table 3 we can observe that the error rate of LPKDA is the same as that of KDA and is relatively high compared with CKFD, KLFDA, and LPP, when the training data size is relative small (e.g., *n*=2). This implies that it is difficult for the proposed LPKDA algorithm to capture more local or global structure information when the training data size is small, thus fusing both local and global structure information does not help. For the results on PIE database listed in Table 5, it is interesting to note that KDA, CKFD, LPP, KLFDA, and LPKDA all achieve comparably low error rates when the training data size is large. Considering the large variance of images in PIE database, this may be due to the fact that in some cases when the training data size and the data variance is large, the useful local geometrical structure information for class classification is corrupted by the densely and randomly distributed sample points, causing LPP techniques to capture no more new information other than global structure information, hence integrating both local and global structure information makes little help in improving performance.

### 4.2. *Experiment on artificial object recognition*

The COIL20 image database [41] contains 1440 images of 20 objects (72 images per subject). The images of each subject were taken every 5 degree apart as the object was rotated on a turntable. Each image is of size $128 \times 128$. Fig.3 shows some examples from the database.

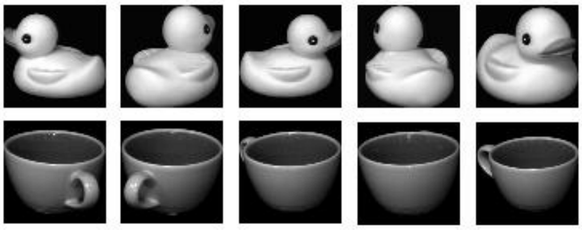


Fig.3. Sample images from COIL20 database.

Table 3. The average error rates (%) across 10 tests and their standard deviations (std) on Yale database.

| Training/Testing numbers | KPCA | KDA | CKFD | LPP | KLFDA | LPKDA |
|---|---|---|---|---|---|---|
| 2/9 | 63.7<br>±4.26 | 55.6<br>±3.15 | 50<br>±2.68 | 44.5<br>±2.68 | 47.4<br>±3.88 | **55.6**<br>**±3.27** |
| 3/8 | 50<br>±4.04 | 37.5<br>±2.95 | 33<br>±2.77 | 35.3<br>±2.68 | 36.1<br>±2.43 | **31.9**<br>**±3.17** |
| 5/6 | 39.24<br>±3.79 | 24.84<br>±3.04 | 21.83 ±<br>2.58 | 24.1<br>±2.13 | 23.75<br>±2.76 | **20.47**<br>**±2.87** |
| 6/5 | 36.4<br>±3.25 | 21.8<br>±2.60 | 18.2<br>±2.33 | 20<br>±2.37 | 21.4<br>±2.54 | **16.4**<br>**±2.67** |

Table 4. The average error rates (%) across 10 tests and their standard deviations (std) on ORL database.

| Training/Testing numbers | KPCA | KDA | CKFD | LPP | KLFDA | LPKDA |
|---|---|---|---|---|---|---|
| 2/8 | 40<br>±3.05 | 26.6<br>±2.32 | 17.8<br>±2.75 | 26.3<br>±3.05 | 26.8<br>±2.75 | **16.3**<br>**±2.13** |
| 3/7 | 29<br>±2.88 | 12.2<br>±2.14 | 11.1<br>±2.15 | 16.8<br>±2.84 | 15.3<br>±2.04 | **7.5**<br>**±2.14** |
| 5/5 | 22.6<br>±2.35 | 6.35<br>±1.76 | 4.34<br>±1.37 | 9.3<br>±1.99 | 9.1<br>±1.81 | **2.75**<br>**±1.18** |
| 6/4 | 21.9<br>±2.38 | 4.4<br>±1.78 | 3.8<br>±1.93 | 7.5<br>±2.04 | 6.6<br>±2.29 | **2.5**<br>**±1.69** |

Table 5. The average error rates (%) across 10 tests and their standard deviations (std) on PIE database.

| Training/Testing numbers | KPCA | KDA | CKFD | LPP | KLFDA | LPKDA |
|---|---|---|---|---|---|---|
| 30/140 | 27.48<br>±0.98 | 10.01<br>±1.01 | 9.08<br>±0.97 | 14.77<br>±0.78 | 13.35<br>±1.65 | **8.29**<br>**±1.12** |
| 60/110 | 23.8<br>±0.88 | 5.5<br>±1.05 | 5.0<br>±0.93 | 6.7<br>±0.69 | 5.37<br>±1.09 | **4.7**<br>**±1.03** |
| 90/80 | 22.3<br>±0.88 | 3.9<br>±0.83 | 3.3<br>±0.82 | 4.1<br>±0.55 | 3.8<br>±0.77 | **3.3**<br>**±0.96** |
| 120/50 | 22<br>±0.69 | 3.2<br>±0.91 | 2.9<br>±0.78 | 3.2<br>±0.54 | 3.1<br>±1.21 | **2.9**<br>**±1.01** |

In our experiments, each image is resized to have a resolution of $64 \times 64$ and 36 samples are randomly chosen from each class for training, while the remaining 36 samples are used for testing. In this way, we run the system 20 times and obtain 10 different training and testing sample sets for both the stages of model selection and performance evaluation.

The error rates of the random 10 different splits on COIL20 database with KPCA, KDA, CKFD, LPP, KLFDA and the proposed LPKDA are presented in Fig.4. The mean error rates and standard deviations of the 10 different partitions are reported in Table 6. From Fig.4 and Table 6, it can be seen that 1) KPCA has the lowest performance among all the tested methods and our proposed LPKDA algorithm consistently outperforms KDA, CKFD, KLFDA, and LPP. 2) Both the local and global geometrical structure information are effective for class classification, and fusing both of them via LPKDA can further improve recognition accuracy.

Table 6. The average error rates (%) across 10 tests and their standard deviations (std) on COIL20 database.

| Methods | KPCA | KDA | CKFD | LPP | KLFDA | LPKDA |
|---|---|---|---|---|---|---|
| Error rates | 25.63<br>±2.22 | 7.85<br>±1.89 | 5.91<br>±1.63 | 8.74<br>±2.2 | 8.6<br>±2.35 | **4.36**<br>**±1.45** |

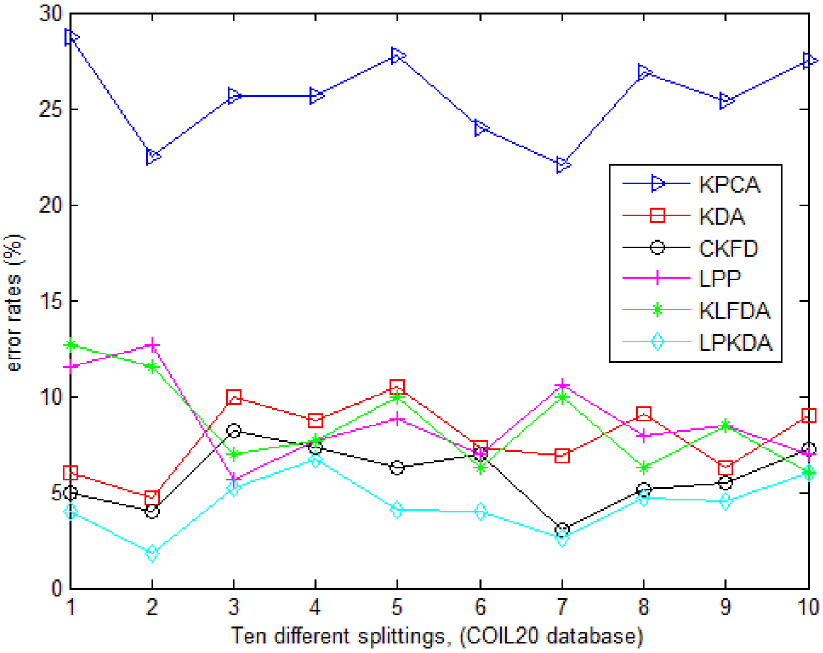


Fig.4. Comparison of KPCA, KDA, CKFD, LPP, KLFDA , and LPKDA in error rates on COIL20 database.

## 5. Conclusion, discussion and future work

In this paper, we have proposed a new feature extraction algorithm, called locality preserving KDA, to integrate both global and local geometrical structure information for feature extraction and classification. The new algorithm first casts KDA as a least squares problem and then uses locality preserving projection as a regularization term to model the local geometrical structure. Extensive experimental results on Yale, ORL, PIE, and COIL20 image databases demonstrate the effectiveness of our approach.

Considering the results listed in Table 5 which show that in some cases when the training data size and the data variance is large, the useful local geometrical structure information for class classification is corrupted by the densely and randomly distributed sample points, it is interesting to think about the possibility of the existence of “support” samples by which useful local geometrical structure information for class classification can be fully determined (hereinafter we call these samples the local-structure-supported vectors, or simply LSS vectors ) and how to locate them. If LSS vectors exist, then by finding them in the training stage, two benefits can be expected: 1) LPP can be efficiently computed since only the LSS vectors are involved in the calculation and most of the "noisy" samples are neglected; 2) with the useful local structure information for classification, the system performance can further be improved.

One of the tested methods, the CKFD algorithm, also achieves relatively good performance in our tests. Since CKFD makes full use of two kinds of discriminant information (regular and irregular, which extracted from the range space and null space of the within-class scatter matrix, respectively) while KDA only uses regular discriminant information, it is also worth to explore the possibility of improving system performance by combing CKFD and the proposed LPKDA.

## Acknowledgements

This work was supported by the China Postdoctoral Science Foundation (Grant No. 2012M511804 ).

## References

1. H. Murase, S.K. Nayar, "Visual learning and recognition of 3-D objects from appearance", International Journal of Computer Vision, vol.14, no.1, pp.5-24, 1995.
2. A.M. Martinez, A.C. Kak, "PCA versus LDA", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 23, no.2, pp.228-233, 2001.
3. D. Xu, S. Yan, D. Tao, S. Lin, H.J. Zhang, "Marginal fisher analysis and its variants for human gait recognition and content-based image retrieval", IEEE Trans. Image Processing, vol.16, no.2, pp.2811-2821, 2007.
4. Jie Gui, ZhenanSun, WeiJia, RongxiangHu, YingkeLei, ShuiwangJi, "Discriminant sparse neighborhood preserving embedding for face recognition", Pattern Recognition, vol.45, no.8, pp.2884-2893, 2012.
5. Quanxue Gao, JingjingLiu, HaijunZhang, JunHou, XiaojingYang, "Enhanced fisher discriminant criterion for image recognition", Pattern Recognition, vol.45, no.10, pp.3717-3724, 2012.
6. B. Scholkopf, A. Smola, and K.R. Muller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem", Neural Computation, vol. 10, no. 5, pp.1299-1319, 1998.
7. S. Mika, G. Ratsch, J. Weston, B. Scholkopf, K.-R.Muller, "Fisher Discriminant Analysis with Kernels", in Proc. IEEE Int'l Workshop Neural Networks for Signal Processing IX, pp. 41-48, Aug, 1999.
8. G. Baudat and F. Anouar, "Generalized Discriminant Analysis Using a Kernel Approach", Neural Computation, vol.12, no.10, pp.2385-2404, 2000.
9. S. Mika, A.J. Smola, and B. Scholkopf, "An Improved Training Algorithm for Kernel Fisher Discriminants", in Proc. Eighth Int'l Workshop Artificial Intelligence and Statistics, T. Jaakkola and T. Richardson, eds., pp. 98-104, 2001.
10. S. Mika, G. Ratsch, J Weston, B. Scholkopf, A. Smola, and K.-R. Muller, "Constructing Descriptive and Discriminative Nonlinear Features: Rayleigh Coefficients in Kernel Feature Spaces", IEEE Trans. Pattern Analysis and Machine Intelligence, vol.25, no.5, pp.623-628, 2003.
11. J. Lu, K.N. Plataniotis, and A.N. Venetsanopoulos, "Face Recognition Using Kernel Direct Discriminant Analysis Algorithms", IEEE Trans. Neural Networks, vol.14, no.1, pp. 117-126, 2003.
12. 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, vol.15, no.2, pp.263-270, 2002.
13. G.C. Cawley and N.L.C. Talbot, "Efficient Leave-One-Out Cross Validation of Kernel Fisher Discriminant Classifiers", Pattern Recognition, vol.36, no.11, pp.2585-2592, 2003.
14. Jian Yang, Alejandro F. Frangi, Jing-yu Yang, David Zhang, and Zhong Jin, "KPCA Plus LDA: A Complete Kernel Fisher Discriminant Framework for Feature Extraction and Recognition", IEEE Trans. Pattern Analysis and Machine Intelligence, vol.27, no.2, pp.230-244, 2005.
15. S.-J.Kim, A.Magnani, and S.Boyd, "Optimal Kernel Selection in Kernel Fisher Discriminant Analysis", in Proc. International Conference on Machine Learning, pp.465-472, 2006.
16. C. Cortes, M. Mohri, A. Rostamizadeh, "Two-stage learning kernel algorithms", in: Proc. the 27th International Conference on Machine Learning, 2010.
17. Yen-Yu Lin, Tyng-Luh Liu, and Chiou-Shann Fuh, "Multiple Kernel Learning for Dimensionality Reduction", IEEE Trans. Pattern Analysis and Machine Intelligence, vol.33, no.6, pp.1147-1160, 2011.
18. S.T. Roweis, L.K. Saul, "Nonlinear dimensionality reduction by locally linear embedding", Science, vol.290, no.5500, pp.2323-2326, 2000.
19. J.B. Tenenbaum, V. de Silva, J.C. Langford, "A global geometric framework for nonlinear dimensionality reduction", Science, vol.290, no.5500, pp.2319-2323, 2000.
20. M. Belkin, P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering", in Advances in Neural Information Processing Systems, vol.1, pp.585–592, 2002.
21. S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, S. Lin, "Graph embedding and extensions: a general framework for dimensionality reduction", IEEE Trans. Pattern

Analysis and Machine Intelligence, vol.29, no.1, pp.40-51, 2007.

22. X. He and P. Niyogi, “Locality preserving projections”, in Advances in Neural Information Processing Systems , 2003.
23. Xiaofei He, Shuicheng Yan, Yuxiao Hu, Partha Niyogi, and Hong-Jiang Zhang, “Face Recognition Using Laplacianfaces”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol.27, no.3, pp.328-340, 2005.
24. W. Yu, X. Teng, C. Liu, “Face recognition using discriminant locality preserving projections”, Image and Vision Computing, vol.24, pp. 239–248, 2006.
25. J. Yang, D. Zhang, J. Yang, B. Niu, “Globally maximizing, locally minimizing: unsupervised discriminant projection with applications to face and palm biometrics”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol.29, no.4, pp. 650–664, 2007.
26. L. Zhu, S. Zhu, “Face recognition based on orthogonal discriminant locality preserving projections”, Neurocomputing, vol.70, pp.1543–1546, 2007.
27. H.-P. Cai, K. Mikolajczyk, J. Matas, “Learning linear discriminant projections for dimensionality reduction of image descriptors”, IEEE Trans. Pattern Analysis and Machine Intelligence vol.33,no.(2), pp. 338–352, 2011.
28. Wankou Yang, ChangyinSun, LeiZhang, “A multi-manifold discriminant analysis method for image feature extraction”, Pattern Recognition, vol.44, no.8, pp. 1649–1657, 2011.
29. W.K. Wong, H.T. Zhao, “Supervised optimal locality preserving projection”, Pattern Recognition, vol.45, no.1, pp. 186–197, 2012.
30. L. Yang, W. Gong, X. Gu, W. Li, Y. Liang, “Null space discriminant locality preserving projections for face recognition”, Neurocomputing, vol.71, pp.3644–3649, 2008.
31. D. Cai, X. He, J. Han, “Spectral regression: a unified approach for sparse subspace learning”, in: Proceedings of the International Conference on Data Mining, 2007.
32. D. Cai, X. He, J. Han, “Spectral Regression: A Unified Subspace Learning Framework for Content-Based Image Retrieval”, ACM Multimedia, Sep. 2007.
33. M. Belkin, P. Niyogi, V. Sindhwani, “Manifold regularization: a geometric framework for learning from labeled and unlabeled examples”, The Journal of Machine Learning Research, vol.7, pp. 2399–2434, 2006.
34. J. Chen, J. Ye, Q. Li, “Integrating global and local structures: a least squares framework for dimensionality reduction”, in: IEEE Conference on Computer Vision and Pattern Recognition, pp.1–8.
35. K. Fukunaga, “Introduction to Statistical Pattern Recognition”, Academic Press, 2nd edition, 1990.
36. B.Scholkopf, A.J.Smola, “Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond”, The MIT Press, 2002.
37. D. Cai, X. He, J. Han, “SRDA: an efficient algorithm for large-scale discriminant analysis”, IEEE Trans. Knowledge and Data Engineering, vol.20, pp.1-12, 2008.
38. U. Von Luxburg, “A tutorial on spectral clustering”, Statistics and Computing, vol.17, pp.395-416, 2007.
39. C. Paige, M. Saunders, “LSQR: an algorithm for sparse linear equations and sparse least squares”, ACM Transactions on Mathematical Software, vol.8, pp.43-71, 1982.
40. C. Paige, M. Saunders, “Algorithm 583 LSQR: sparse linear equations and least squares problems”, ACM Transactions on Mathematical Software, vol.8, pp.195-209, 1982.
41. COIL20imagedatabase, http://www1.cs.columbia.edu/ CAVE/software/soft lib/coil-20.phpS.
42. Yale Univ. Face Database, http://cvc.yale.edu/projects/ yalefacesyalefaces. htmlS.
43. The ORL database of faces, http://www.cl.cam.ac.uk/ Research/DTG/.
44. T. Sim, S. Baker, M. Bsat, “The CMU Pose, Illumination, and Expression (PIE) Database”, in: Proc. IEEE Int’l Conf. Automatic Face and Gesture Recognition, May 2002.
45. M.H. Yang, “Kernel Eigenfaces vs. Kernel Fisherfaces: Face Recognition Using Kernel Methods,” in: Proc. Fifth IEEE Int’l Conf. Automatic Face and Gesture Recognition, pp. 215-220, May 2002.
46. K.-R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, “An Introduction to Kernel-Based Learning Algorithms,” IEEE Trans. Neural Networks, vol.12, no.2, pp.181-201, 2001.
47. Masashi Sugiyama, “Local Fisher Discriminant Analysis for Supervised Dimensionality Reduction”, in: Proc. The 23rd international conference on Machine learning, pp. 905-912, 2006.