Open access • Proceedings Article • DOI:10.1109/ICPR.2008.4761132

Ridge Regression for Two Dimensional Locality Preserving Projection

Nam Nguyen, Wanquan Liu, Svetha Venkatesh

Institutions: Curtin University

Published on: 01 Jan 2008 - International Conference on Pattern Recognition

Topics: Projection (set theory) and FERET database

Ridge Regression for Two Dimensional Locality Preserving Projection

Nam Nguyen, Wanquan Liu and Svetha Venkatesh
Department of Computing, Curtin University)
{Thanh.Nguyen,W.Liu,S.Venkatesh}@curtin.edu.au

Abstract

Two Dimensional Locality Preserving Projection (2D-LPP) is a recent extension of LPP, a popular face recognition algorithm. It has been shown that 2D-LPP performs better than PCA, 2D-PCA and LPP. However, the computational cost of 2D-LPP is high. This paper proposes a novel algorithm called Ridge Regression for Two Dimensional Locality Preserving Projection (RR-2DLPP), which is an extension of 2D-LPP with the use of ridge regression. RR-2DLPP is comparable to 2D-LPP in performance whilst having a lower computational cost. The experimental results on three benchmark face data sets – the ORL, Yale and FERET databases – demonstrate the effectiveness and efficiency of RR-2DLPP compared with other face recognition algorithms such as PCA, LPP, SR, 2D-PCA and 2D-LPP.

1 Introduction

Face recognition has attracted much research effort in recent years. Well-known algorithms in face recognition include Eigenface [14] and Fisherface [1]. Eigenface is an unsupervised method based on Principal Component Analysis (PCA) [6], which uses the Karhunen-Loeve transform to project images onto a lower dimension subspace for maximizing the variance of training images. Extensions of Eigenface include Nonlinear PCA [10] and Kernel PCA [17]. Fisherface is a supervised method based on PCA and Linear Discriminant Analysis (LDA) [6], and intends to find a linear transform to maximize the ratio of the between-class and within-class distances. Extensions of Fisherface include Kernel Fisherface [16] and Null Space LDA [4].

All the above algorithms work on a vector representation of images and need to compute the eigenvectors of a high-dimensional covariance matrix in order to find the optimal linear transformation. When the size of the image is large, these algorithms may have computing problems in eigen-decomposition. To avoid this, a few algorithms have been proposed to work directly on matrix representation of images such as Two Dimensional PCA (2D-PCA) [15], 2D-LDA [18] and 2D-LPP [9]. 2D-LPP has been shown to be extension of 2D-PCA and 2D-LDA [9]. However, the computational cost of 2D-LPP is high because it involves dense matrix eigen-decomposition and operations on Kronecker products of matrices

In this paper, we propose a novel algorithm called Ridge Regression for Two Dimensional Locality Preserving Projections (RR-2DLPP). RR-2DLPP is an extension of 2D-LPP with the use of ridge regression [13]. The motivation for RR-2DLPP is from Cai *et al.* [3], who demonstrate the advantages of ridge regression in boosting the recognition accuracy in the case of vectorized images. For 2D images, direct combination of 2D-LPP and ridge regression involves eigen-decomposition of the Kronecker products of high-dimensional matrices and is computationally expensive. We propose two theorems to help RR-2DLPP avoid dense matrix eigen-decompositions and operations on Kronecker products of matrices, making RR-2DLPP less computationally expensive than 2D-LPP. We conducted experiments on three benchmark face data sets: ORL, Yale, and FERET databases. The experimental results show that RR-2DLPP is comparable not only to 2D-LPP but also other face recognition algorithms such as PCA, LPP, SR [3] and 2D-PCA.

The contributions of this paper are as follows: (1) proposal of a novel framework, which integrates ridge regression into 2D-LPP, (2) proposal of two theorems on relations among eigenvalues and eigenvectors of generalized eigenvalues problems, (3) application of these theorems in a new framework to reduce the computational cost, and (4) extensive experiments to demonstrate the effectiveness and efficiency of the proposed framework.

The remaining of the paper is structured as follows. Section 2 describes 2D-LPP. The proposed algorithm, RR-2DLPP, is presented in Section 3. The experimental results are shown in Section 4, followed by some conclusion remarks in Section 5.

2 2D-LPP

2D-LPP was proposed by Hu *et al.* [9] as an extension of LPP [8] that works directly on 2D images. Assume that $\mathbf{X}_1, \ldots, \mathbf{X}_m$ are the matrix representations of training images and $\mathbf{X}_i \in \mathbb{R}^{n_1 \times n_2}$ $(\forall i = 1, \ldots, m)$.

2D-LPP aims to find an optimal matrix $\mathbf{A} \in \mathbb{R}^{n_2 \times d}$ to project a face image $\mathbf{X}$ to $f(\mathbf{X}) = \mathbf{XA} \in \mathbb{R}^{n_1 \times d}$, where d is the reduced width of the image ($d \leq n_2$). Denote $\mathbf{W} = \{w_{ij}\}_{m \times m}$ is the similarity matrix defined by $\mathbf{X}_1, \ldots, \mathbf{X}_m$. 2D-LPP aims to preserve the similarity matrix in the projection space by solving the following optimization problem

$$\mathbf{A} = \arg\min_{\mathbf{A}} \sum_{i,j}^{m} \| f(\mathbf{X}_i) - f(\mathbf{X}_j) \|_F^2 \times w_{ij} \quad (1)$$

where $\| \,.\, \|_F$ is Frobenius norm [7], and a suitable constraint is applied to $f(\mathbf{X}_i)$ in order to remove an arbitrary scaling factor.

Let $\mathbf{D} = \{d_{ij}\}_{m \times m}$, where $d_{ij} = 0$ if $i \neq j$, otherwise $d_{ii} = \sum_j w_{ij}$. Also define $\mathbf{L} = \mathbf{D} - \mathbf{W}$, which is the Laplacian of the graph formed by $\mathbf{X}_1, \ldots, \mathbf{X}_m$ [5]. From Eq. 1 and according to derivations in [9], we have

$$\mathbf{A} = \arg\min_{\mathbf{A}}(\text{trace}(\mathbf{A}^T \mathbb{X}^T (\mathbf{L} \otimes \mathbf{I}_{n_1}) \mathbb{X} \mathbf{A})) \quad (2)$$

where $\mathbb{X}$ is an $(m \times n_1) \times n_2$ matrix generated by arranging $\mathbf{X}_1, \ldots, \mathbf{X}_m$ in column, $\mathbf{I}_{n_1}$ is an $n_1 \times n_1$ identity matrix, and $\mathbf{L} \otimes \mathbf{I}_{n_1}$ is the Kronecker product of $\mathbf{L}$ and $\mathbf{I}_{n_1}$. Choosing the constraint for $f(\mathbf{X})$ as $\mathbf{A}^T \mathbb{X}^T (\mathbf{D} \otimes \mathbf{I}_{n_1}) \mathbb{X} \mathbf{A} = \mathbf{I}_d$ and replacing $\mathbf{L} = \mathbf{D} - \mathbf{W}$ in Eq. 2, we have

$$\mathbf{A} = \arg\max_{\mathbf{A}} \frac{\text{trace}(\mathbf{A}^T \mathbb{X}^T (\mathbf{W} \otimes \mathbf{I}_{n_1}) \mathbb{X} \mathbf{A})}{\text{trace}(\mathbf{A}^T \mathbb{X}^T (\mathbf{D} \otimes \mathbf{I}_{n_1}) \mathbb{X} \mathbf{A})} \quad (3)$$

The solution $\mathbf{A}$ for the above optimization problem can be obtained by solving the following generalized eigenvalue (GE) problem

$$\mathbb{X}^T (\mathbf{W} \otimes \mathbf{I}_{n_1}) \mathbb{X} \mathbf{A} = \lambda \mathbb{X}^T (\mathbf{D} \otimes \mathbf{I}_{n_1}) \mathbb{X} \mathbf{A} \quad (4)$$

Theoretically, 2D-LPP first computes $\mathbb{X}^T (\mathbf{W} \otimes \mathbf{I}_{n_1}) \mathbb{X}$ and $\mathbb{X}^T (\mathbf{D} \otimes \mathbf{I}_{n_1}) \mathbb{X}$, then chooses $\mathbf{A}$ as the eigenvectors associated with the largest eigenvalues λ in Eq. 4. In practice, there are different ways of choosing the similarity matrix $\mathbf{W}$. In what follows, $\mathbf{W}$ is chosen as: $w_{ij} = \| \mathbf{X}_i - \mathbf{X}_j \|_F^2$ if $\mathbf{X}_j$ is among the k nearest neighbors of $\mathbf{X}_i$, otherwise $w_{ij} = 0$.

The disadvantage of 2D-LPP is its high computational complexity. The main computational cost of 2D-LPP is due to the calculation of $\mathbb{X}^T (\mathbf{W} \otimes \mathbf{I}_{n_1}) \mathbb{X}$ and $\mathbb{X}^T (\mathbf{D} \otimes \mathbf{I}_{n_1}) \mathbb{X}$, which has computational complexity of $\mathcal{O}(n^4 m^3)$, where $n = \max(n_1, n_2)$.

3 Ridge Regression for 2D-LPP

RR-2DLPP is an extension of 2D-LPP with the use of ridge regression [13]. The motivation for RR-2DLPP is that we want to solve the GE problem in Eq. 4 via a regression technique; thus we can avoid the computation of $\mathbb{X}^T (\mathbf{W} \otimes \mathbf{I}_{n_1}) \mathbb{X}$ and $\mathbb{X}^T (\mathbf{D} \otimes \mathbf{I}_{n_1}) \mathbb{X}$, which are computationally expensive. In order to achieve this reduction in computation, we make use of the following theorem:

Theorem 1: *Assume that λ is an eigenvalue and $\mathbf{y}$ is the corresponding eigenvector of the GE problem*

$$(\mathbf{W} \otimes \mathbf{I}_{n_1}) \mathbf{y} = \lambda (\mathbf{D} \otimes \mathbf{I}_{n_1}) \mathbf{y}, \ \lambda \neq 0 \quad (5)$$

If $\mathbf{y} = \mathbb{X}\mathbf{a}$, then λ and $\mathbf{a}$ will be the eigenvalue and corresponding eigenvector of the GE problem in Eq. 4.

Proof: Actually, one can derive

$$\mathbb{X}^T (\mathbf{W} \otimes \mathbf{I}_{n_1}) \mathbb{X} \mathbf{a} = \mathbb{X}^T (\mathbf{W} \otimes \mathbf{I}_{n_1}) \mathbf{y} = \mathbb{X}^T \lambda (\mathbf{D} \otimes \mathbf{I}_{n_1}) \mathbf{y}$$
$$= \lambda \mathbb{X}^T (\mathbf{D} \otimes \mathbf{I}_{n_1}) \mathbb{X} \mathbf{a}$$

Thus, λ is an eigenvalue and $\mathbf{a}$ is an corresponding eigenvector of the GE problem in Eq. 4■

By Theorem 1, instead of solving the GE problem in Eq. 4, we can solve the GE problem in Eq. 5, then find $\mathbf{a}$ such that $\mathbf{y} = \mathbb{X}\mathbf{a}$. However, directly solving the GE problem in Eq. 5 is still computationally expensive due to the eigen-decomposition of a high-dimensional matrix. The following theorem helps us reduce the cost of this task.

Theorem 2: *Assume that λ is the eigenvalue and $\mathbf{z}$ is the corresponding eigenvector of the GE problem*

$$\mathbf{Wz} = \lambda \mathbf{Dz}, \ \lambda \neq 0 \quad (6)$$

Let $\mathbf{v}$ be any non-zero unit vector in $\mathbb{R}^{n_1}$ and $\mathbf{y} = \mathbf{z} \otimes \mathbf{v}$, then λ and $\mathbf{y}$ are the eigenvalue and corresponding eigenvector of the GE problem in Eq. 5.

Proof: One can observe that

$$(\mathbf{W} \otimes \mathbf{I}_{n_1}) \mathbf{y} = (\mathbf{W} \otimes \mathbf{I}_{n_1})(\mathbf{z} \otimes \mathbf{v}) = (\mathbf{Wz}) \otimes (\mathbf{I}_{n_1} \mathbf{v})$$
$$= \lambda (\mathbf{Dz}) \otimes (\mathbf{I}_{n_1} \mathbf{v}) = \lambda (\mathbf{D} \otimes \mathbf{I}_{n_1})(\mathbf{z} \otimes \mathbf{v}) = \lambda (\mathbf{D} \otimes \mathbf{I}_{n_1}) \mathbf{y}$$

Thus, λ is an eigenvalue and $\mathbf{y}$ is an corresponding eigenvector of the GE problem in Eq. 5■

By Theorem 2, instead of solving the GE problem in Eq. 5, we can solve the GE problem in Eq. 6. We can obtain λ and $\mathbf{z}$ from Eq. 6 with lower computational cost because $\mathbf{D}$ is a diagonal matrix and the sizes of $\mathbf{D}$ and $\mathbf{W}$ are smaller than the matrices in Eq. 4 and 5.

Based on the above observations, we propose RR-2DLPP algorithm, in which the transformation matrix $\mathbf{A}$ is obtained as follows:

1. Solve Eq. 6 to obtain the largest eigenvalue λ and the corresponding eigenvector $\mathbf{z}$.

2. Select $\mathbf{v}_1, \ldots, \mathbf{v}_d$ are d mutually orthogonal unit vectors in space $\mathbb{R}^{n_1}$. There are many ways to select $\mathbf{v}_1, \ldots, \mathbf{v}_d$ satisfied that condition. In practice, $\mathbf{v}_1, \ldots, \mathbf{v}_d$ are defined as follows. First, we select $\mathbf{v}_1 = [1, 1, \ldots, 1]^T$, $\mathbf{v}_i = [0, 0, \ldots, 0, 1, 0, \ldots, 0]^T$ for all $2 \leq i \leq d$ (all elements of $\mathbf{v}_i$ are zero, except the i^{th} element is one). Then, the Gram-Schmidt process is used to orthogonalize $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_d$.

Algorithm 1 RR-2DLPP

Input: m training images $\mathbf{X}_1, \ldots, \mathbf{X}_m \in \mathbb{R}^{n_1 \times n_2}$, reduced width d , regularized parameters α.

Output: transformation matrix $\mathbf{A}$, $f(\mathbf{X}_1), \ldots, f(\mathbf{X}_m)$.

Algorithm:

1. Compute the similarity matrix $\mathbf{W}$ of $\mathbf{X}_1, \ldots, \mathbf{X}_m$. Let $\mathbf{D} = \{d_{ij}\}_{m \times m}$, where $d_{ij} = 0$ if $i \neq j$, otherwise $d_{ii} = \sum_j w_{ij}$.
2. Obtain the largest eigenvalue λ and the corresponding eigenvector $\mathbf{z}$ from Eq. 6.
3. Select $\mathbf{v}_1, \ldots, \mathbf{v}_d$ being d mutually orthogonal unit vectors in $\mathbb{R}^{n_1}$.
4. Obtain $\mathbf{y}_1, \ldots, \mathbf{y}_d$ as $\mathbf{y}_i = \mathbf{z} \otimes \mathbf{v}_i$ $(i = 1, \ldots, d)$.
5. Let $\mathbb{Y} = [\mathbf{y}_1\ \mathbf{y}_2 \ldots\ \mathbf{y}_d]$ and $\mathbb{X} = [\mathbf{X}_1^T\ \mathbf{X}_2^T \ldots\ \mathbf{X}_m^T]^T$.
6. Obtain $\mathbf{A}$ from Eq. 8 and compute $f(\mathbf{X}_i) = \mathbf{X}_i\mathbf{A}$, $\forall i = 1, \ldots, m$.

Testing: A test image $\mathbf{X}$ is matched to training images as follows:

1. Compute: $f(\mathbf{X})$=$\mathbf{XA}$.
2. Compare $f(\mathbf{X})$ with $f(\mathbf{X}_1), \ldots, f(\mathbf{X}_m)$ to find the best match of $f(\mathbf{X})$ using the nearest neighborhood classifier.

3. Obtain $\mathbf{y}_1, \ldots, \mathbf{y}_d$ as $\mathbf{y}_i = \mathbf{z} \otimes \mathbf{v}_i$ $(i = 1, \ldots, d)$. By Theorem 2, $\mathbf{y}_1, \ldots, \mathbf{y}_d$ are eigenvectors corresponding to eigenvalue λ in Eq. 5.
4. Let $\mathbb{Y} = [\mathbf{y}_1\ \mathbf{y}_2 \ldots\ \mathbf{y}_d]$. We want to obtain $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2 \ldots\ \mathbf{a}_d]$ such that $\mathbb{X}\mathbf{A} = \mathbb{Y}$, and thus by Theorem 1, each column $\mathbf{a}_i$ of $\mathbf{A}$ will be an eigenvector corresponding to eigenvalue λ in Eq. 4. In reality, such $\mathbf{A}$ might not exist. A possible way is to find $\mathbf{A}$ from the following ridge regression problem

$$\mathbf{A} = \arg\min_{\mathbf{A}}\Big(\sum_{i=1}^{m \times n_1} \parallel \bar{\mathbf{x}}_i\mathbf{A} - \bar{\mathbf{y}}_i \parallel_2^2 + \alpha \parallel \mathbf{A} \parallel_F^2\Big) \tag{7}$$

where $\bar{\mathbf{x}}_i$ is the i^{th} row of $\mathbb{X}$, $\bar{\mathbf{y}}_i$ is the i^{th} row of $\mathbb{Y}$, and α is the regularization parameter. We can derive from Eq. 7 that

$$\mathbf{A} = (\mathbb{X}^T\mathbb{X} + \alpha\mathbf{I}_{n_2})^{-1}\mathbb{X}^T\mathbb{Y} \tag{8}$$

Details of RR-2DLPP are shown in Algorithm 1. The main computational cost of RR-2DLPP is due to the calculation of $\mathbf{A}$ in Eq. 8. RR-2DLPP has computational complexity of $\mathcal{O}(n^4m^2)$, which is significantly less expensive than 2D-LPP with computational complexity of $\mathcal{O}(n^4m^3)$.

4 Experimental results

We undertook experiments on the following three benchmark face data sets: ORL, Yale and FERET databases. The training images were unlabeled in all experiments. Our proposed algorithm, RR-2DLPP, was compared with 2D-LPP [9] and other well-known algorithms: PCA [14], LPP [8], SR [3] and 2D-PCA [15]. RR-2DLPP was run with the LPP neighborhood size $k = 5$, dimension reduction parameter $d = 15$, and regularization parameter $\alpha = 0.01$. These values were chosen by practice, and used for all face database.

Table 1. Recognition accuracy (%) of RR-2DLPP on the ORL database

	1-train	2-train	3-train	4-train	5-train
PCA [14]	65.6±2.6	80.0±2.5	86.0±2.4	88.8±2.2	92.0±1.5
LPP [8]	57.0RR-2DLPP±3.6	71.6±3.3	78.4±3.6	82.0±3.0	87.4±2.5
SR [3]	60.0±2.8	74.6±3.1	82.0±3.0	86.9±2.6	91.6±1.8
2D-PCA [15]	67.1±3.2	80.6±2.8	86.9±2.4	89.7±2.4	92.5±1.6
2D-LPP [9]	70.1±2.8	83.3±2.5	88.3±1.9	91.7±1.6	94.2±1.1
RR-2DLPP	74.4±2.5	85.5±2.6	89.5±2.3	91.9±2.0	94.4±1.2

4.1 Experiments on the ORL

The ORL database[1] has 400 face images of 40 people, each has 10 face images. The images were resized to 64×64. Five experiments (1-train, 2-train, ..., 5-train) were considered, where $i-$train experiment corresponds to i images of each person being used for training and the remaining images for testing. For each experiment, 20 random splits (train images, test images) of the ORL database were created. We ran RR-2DLPP on these random splits of the ORL database and average the results. The experiments were conducted on a Pentium 4 3.2GHz Desktop PC.

Table 1 shows the recognition accuracy of RR-2DLPP compared with the top recognition accuracy of PCA, LPP, SR, 2D-PCA and 2D-LPP. The results show that RR-2DLPP is superior to the other algorithms in all experiments. Table 2 shows the training time of RR-2DLPP compared with PCA, LPP and 2D-LPP. The table shows that RR-2DLPP is much faster than LPP and 2D-LPP in all experiments. Although 2D-LPP is not faster than the PCA and SR in the 1-train and 2-train experiments, it is faster than these algorithms in the 3-train, 4-train and 5-train experiments.

4.2 Experiments on the Yale face database

The Yale face database[2] has 165 face images of 15 people with each person having 11 images. These images were resized to 64×64. We again tested the performance of RR-2DLPP in five experiments: 1-train, 2-train,. . . , 5-train. In each $i-$train experiment, RR-2DLPP was tested

[1] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html
[2] http://vismod.media.mit.edu/vismod/classes/mas622-00/datasets/

Table 2. Computational cost (in second) of RR-2DLPP on the ORL database.

	1-train	2-train	3-train	4-train	5-train
PCA	0.026	0.071	0.136	0.213	0.317
LPP	0.045	0.095	0.162	0.268	0.347
SR	0.043	0.084	0.139	0.241	0.309
2D-PCA	0.019	0.033	0.047	0.063	0.073
2D-LPP	0.100	0.156	0.241	0.331	0.386
RR-2DLPP ($d = 15$)	0.044	0.084	0.123	0.193	0.229

Table 3. Recognition accuracy (%) of RR-2DLPP on the Yale database.

	1-train	2-train	3-train	4-train	5-train
PCA	60.4±7.4	77.6±3.0	80.1±3.3	81.1±2.2	83.1±3.1
LPP	53.0±8.9	74.4±4.4	79.5±3.7	83.3±2.4	87.7±2.9
SR	60.4±7.5	79.5±3.2	80.1±3.5	82.6±2.0	84.9±2.9
2D-PCA	62.9±7.1	80.5±3.2	82.8±3.3	83.5±2.5	86.7±3.5
2D-LPP	65.8±5.4	80.9±3.9	84.3±3.8	86.6±3.3	88.1±4.0
RR-2DLPP	70.0±5.3	83.3±3.0	85.9±2.9	87.7±2.4	88.7±2.2

on 20 random splits of the Yale database. Table 3 shows the recognition accuracy of RR-2DLPP compared with the top recognition accuracy of other algorithms. The table shows that in all experiments RR-2DLPP performs better than PCA, LPP, SR, 2D-PCA and 2D-LPP.

4.3 Experiments on the FERET database

We also used the FERET database [11, 12] to test the performance of RR-2DLPP. We selected people in FERET having at least four frontal images as in [2]. In total, 1433 images of 240 people were selected. The images were pre-processed using the CSU Face Identification Evaluation System [2], then resized to 64×64. We considered three experiments: 1-train, 2-train and 3-train. For each experiment, 20 random splits (training images, test images) of the database were created. Table 4 shows the recognition accuracy of RR-2DLPP compared with the top recognition accuracy of other algorithms. One can observe that RR-2DLPP is comparable to PCA, LPP, SR, 2D-PCA and 2D-LPP in all experiments.

5 Conclusions

In this paper we have presented a novel algorithm for face recognition, Ridge Regression for Two Dimensional Locality Preserving Projection (RR-2DLPP), which is an extension of 2D-LPP with the use of ridge regression. The recognition accuracy of RR-2DLPP is comparable to 2D-LPP whilst RR-2DLPP have a lower computational cost. Experimental results on the ORL, Yale and FERET databases demonstrate the effectiveness and efficiency of RR-2DLPP compared with other unsupervised face recognition algorithms such as PCA, LPP, SR, 2D-PCA and 2D-LPP.

Table 4. Recognition accuracy (%) of RR-2DLPP on the FERET face database.

	1-train	2-train	3-train
PCA	56.9±1.9	71.6±1.5	80.2±2.0
LPP	44.6±2.4	59.7±1.4	71.4±2.0
SR	44.7±2.0	59.9±1.8	71.3±2.2
2D-PCA	57.5±1.8	72.1±1.5	80.6±1.8
2D-LPP	48.1±1.4	64.5±1.6	79.1±2.2
RR-2DLPP	53.7±1.3	68.6±1.9	77.7±2.0

References

[1] P. Belhumeur, J. Hespanda, and D. Kiregeman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[2] J. Beveridge, D. Bolme, B. Draper, and M. Teixeira. The CSU face identification evaluation system: Its purpose, features, and structure. *Machine Vision and Applications*, 16(2):128–138, 2005.

[3] D. Cai, X. He, and J. Han. Spectral regression for efficient regularized subspace learning. In *IEEE International Conference on Computer Vision (ICCV)*, Rio de Janeiro, Brazil, 2007.

[4] L. Chen, H. Liao, M. Ko, J. Liin, and G. Yu. A new LDA-based face recognition system which can solve the small sample size problem. *Pattern recognition*, 33(10):1713–1726, 2000.

[5] F. Chung. Spectral graph theory. *Regional Conference Series in Mathematics*, 92, 1997.

[6] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley-Interscience, Hoboken, NJ, 2nd edition, date 2000.

[7] G. Golub and C. V. Loan. *Matrix computations*. The Johns Hopkins University Press, MD, USA, 3rd edition edition, date 1996.

[8] Z. He and P. Niyogi. Locality preserving projections. In *NIPS*, volume 16, 2003.

[9] D. Hu, G. Feng, and Z. Zhou. Two-dimensional locality preserving projections (2DLPP) with its application to palmprint recognition. *Pattern Recognition*, 40(1):339–342, 2007.

[10] M. A. Kramer. Nonlinear Principle Component Analysis using autoassociative Neural networks. *American Institution Chemical Engineering Journal*, 32(2), 1991.

[11] P. Phillips, H. Moon, S. Rizvi, and P. Rauss. The FERET evaluation methodology for face recognition algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22:1090–1104, 2000.

[12] P. Phillips, H. Wechsler, J. Huang, and P. Rauss. The FERET database and evaluation procedure for face recognition algorithms. *Image and Vision Computing J*, 16(5):295–306, 1998.

[13] A. Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial Mathematics, date 2005.

[14] M. Turk and A. Pentland. Eigenfaces for recognition. *Cognitive Neuroscience*, 3(1):71–86, 1991.

[15] J. Yang, D. Zhang, A. F. Frangi, and J. Yang. Two-dimensional PCA: A new approach to appearance-based face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):131–137, 2004.

[16] M. Yang. Kernel Eigenfaces vs. kernel Fisherfaces: Face recognition using kernel methods. In *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, DC, USA, 2002. IEEE Computer Society Washington.

[17] M. Yang, N. Ahuja, and D. Kriegman. Face recognition using kernel Eigenfaces. In *In Proceedings of International Conference of Image Processing*, volume 1, pages 37–40, 2000.

[18] J. Ye, R. Janardan, and W. Li. Two-dimensional linear discriminant analysis. In *Advances in Neural Information Processing Systems*, volume 17, 2004.