Neighborhood Preserving Embedding

Xiaofei He[1] Deng Cai[2] Shuicheng Yan[3] Hong-Jiang Zhang[4]
[1] Department of Computer Science, University of Chicago, Chicago, IL 60637
[2] Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61081
[3] Department of Information Engineering, Chinese University of Hong Kong, Hong Kong
[4] Microsoft Research Asia, Beijing, P.R. China
Contact: xiaofei@cs.uchicago.edu

Abstract

Recently there has been a lot of interest in geometrically motivated approaches to data analysis in high dimensional spaces. We consider the case where data is drawn from sampling a probability distribution that has support on or near a submanifold of Euclidean space. In this paper, we propose a novel subspace learning algorithm called Neighborhood Preserving Embedding (NPE). Different from Principal Component Analysis (PCA) which aims at preserving the global Euclidean structure, NPE aims at preserving the local neighborhood structure on the data manifold. Therefore, NPE is less sensitive to outliers than PCA. Also, comparing to the recently proposed manifold learning algorithms such as Isomap and Locally Linear Embedding, NPE is defined everywhere, rather than only on the training data points. Furthermore, NPE may be conducted in the original space or in the reproducing kernel Hilbert space into which data points are mapped. This gives rise to kernel NPE. Several experiments on face database demonstrate the effectiveness of our algorithm.

1. Introduction

Real data of natural and social sciences is often very high-dimensional. However, the underlying structure can in many cases be characterized by a small number of parameters. Reducing the dimensionality of such data is beneficial for visualizing the intrinsic structure and it is also an important preprocessing step in many statistical pattern recognition problems.

In recent years, computer vision research has witnessed a growing interest in discovering the manifold of perceptual observation, [6], [7], [13], [11]. For example, an image can be identified with a point in an abstract *image space*. Typically the image space is a very high dimensional space. The perceptually meaningful structure of these images, however, is of much lower dimensionality. A perceptual system that discovers this manifold structures will support a wide range of recognition, classification, and imagery tasks, despite the absence of any prior physical knowledge about three-dimensional object geometry, surface texture, or illumination conditions.

Learning a manifold of perceptual observation is difficult because these observations usually exhibit significant nonlinear structure. Classical techniques for manifold learning, such as PCA is designed to operate when the manifold is embedded *linearly* or almost linearly in the ambient space. When the class information is available, Linear Discriminant Analysis (LDA) [4] can be used to find a linear subspace which is optimal for discrimination. Meanwhile several nonlinear techniques have been proposed to discover the *nonlinear* structure of the manifold such as Laplacian Eigenmap [2], Locally Linear Embedding (LLE) [11], and Isomap [13]. These nonlinear methods do yield impressive results on some benchmark artificial data sets besides some real applications. However, their nonlinear property makes them computationally expensive. Moreover, they yield mappings that are defined only on the $training$ data points and it remains unclear how to $naturally$ evaluate the maps on novel $testing$ points.

Kernel based techniques, such as kernel PCA [12] and kernel LDA [8] that generate nonlinear maps have also been considered. Most of these methods do not explicitly consider the structure of the manifold on which the data may possibly reside.

In this paper, we propose a new *linear* dimensionality reduction algorithm, called *Neighborhood Preserving Embedding* (NPE). Different from PCA which aims at preserving the global Euclidean structure, NPE aims at preserving the local manifold structure. Given a set of data points in the ambient space, we first build a weight matrix which describes the relationship between the data points. Specifically, for each data point, it is represented as a linear combination of the neighboring data points and the combination coefficients are specified in the weight matrix. We then find an optimal embedding such that the neighborhood structure can be preserved in the dimensionality reduced space.

It is worthwhile to highlight several aspects of the pro-

Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)
1550-5499/05 $20.00 © 2005 **IEEE**

posed approach here:

1. NPE shares some similar properties with the Locality Preserving Projection (LPP) algorithm [5]. Both of them aims to discover the local structure of the data manifold. However, their objective functions are totally different.

2. NPE is linear. This makes it fast and suitable for practical applications. It may be conducted in the original space or in the reproducing kernel Hilbert space (RKHS) into which data points are mapped. This gives rise to kernel NPE.

3. NPE can be performed in either supervised or unsupervised mode. When the class information is available, it can be utilized to build a better weight matrix.

The rest of this paper is organized as follows. The NPE algorithm is proposed in Section 2 followed by a justification in Section 3. Experimental results are shown in Section 4. Finally, we provide some Concluding remarks and suggestions for future work in Section 5.

2. Linear Techniques for Dimensionality Reduction

It is generally believed that the face space is a submanifold embedded in the ambient image space. Two of the most popular linear techniques for learning such a face manifold are Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) [4]. PCA is unsupervised while LDA is supervised.

The basic idea of PCA is to project the data along the directions of maximal variances so that the reconstruction error can be minimized. Given a set of data points $\mathbf{x}_1, \cdots, \mathbf{x}_n$, let $\mathbf{a}$ be the transformation vector and $y_i = \mathbf{a}^T\mathbf{x}_i$. The objective function of PCA is as follows:

$$\begin{aligned}\mathbf{a}_{opt} &= \arg\max_{\mathbf{a}} \sum_{i=1}^{n} (y_i - \overline{y})^2 \\ &= \arg\max_{\mathbf{a}} \mathbf{a}^T C \mathbf{a}\end{aligned}$$

where $\overline{y} = \frac{1}{n}\sum y_i$ and C is the data covariance matrix. The basis functions of PCA are the eigenvectors of the data covariance matrix corresponding to the largest eigenvalues.

While PCA seeks directions that are efficient for representation, LDA seeks directions that are efficient for discrimination. Suppose the data points belong to l classes. The objective function of LDA is as follows:

$$\mathbf{a}_{opt} = \arg\max_{\mathbf{a}} \frac{\mathbf{a}^T S_B \mathbf{a}}{\mathbf{a}^T S_W \mathbf{a}}$$

$$S_B = \sum_{i=1}^{l} n_i \left(\mathbf{m}^{(i)} - \mathbf{m}\right)\left(\mathbf{m}^{(i)} - \mathbf{m}\right)^T$$

$$S_W = \sum_{i=1}^{l} \left(\sum_{j=1}^{n_i} \left(\mathbf{x}_j^{(i)} - \mathbf{m}^{(i)}\right)\left(\mathbf{x}_j^{(i)} - \mathbf{m}^{(i)}\right)^T\right)$$

where $\mathbf{m}$ is the total sample mean vector, n_i is the number of samples in the ith class, $\mathbf{m}^{(i)}$ is the average vector of the ith class, and $\mathbf{x}_j^{(i)}$ is the jth sample in the ith class. We call S_W the within-class scatter matrix and S_B the between-class scatter matrix.

3. Neighborhood Preserving Embedding (NPE)

In this Section, we introduce a new linear dimensionality reduction algorithm, called *Neighborhood Preserving Embedding* (NPE). NPE is a linear approximation to the LLE [11] algorithm. The detailed theoretical justification of our algorithm will be provided in the next Section.

3.1. The linear dimensionality reduction problem

The generic problem of linear dimensionality reduction is the following. Given a set of points $\mathbf{x}_1, \cdots, \mathbf{x}_m$ in $\mathbb{R}^n$, find a transformation matrix A that maps these m points to a set of points $\mathbf{y}_1, \cdots, \mathbf{y}_m$ in $\mathbb{R}^d$ $(d \ll n)$, such that $\mathbf{y}_i$ "represents" $\mathbf{x}_i$, where $\mathbf{y}_i = A^T\mathbf{x}_i$. Our method is of particular applicability in the special case where $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m \in \mathcal{M}$ and $\mathcal{M}$ is a nonlinear manifold embedded in $\mathbb{R}^n$.

3.2. The algorithm

The algorithmic procedure is formally stated below:

1. **Constructing an adjacency graph**: Let G denote a graph with m nodes. The i-th node corresponds to the data point $\mathbf{x}_i$. There are two ways to construct the adjacency graph:

   - *K nearest neighbors (KNN):* Put a *directed* edge from node i to j if $\mathbf{x}_j$ is among the K nearest neighbors of $\mathbf{x}_i$.
   - *ϵ neighborhood:* Put an edge between nodes i and j if $\|\mathbf{x}_j - \mathbf{x}_i\| \leq \epsilon$.

   The graph constructed by the first method is a directed graph, while the one constructed by the second method is an undirected graph. In many real world applications, it is difficult to choose a good ϵ. In this work, we adopt the KNN method to construct the adjacency graph. When computational complexity is a major concern, one may switch to ϵ neighborhood. We denote by $i \sim j$ that there is an edge from i to j.

2. **Computing the weights**: In this step, we compute the weights on the edges. Let W denote the weight matrix with W_{ij} having the weight of the edge from node i to node j, and 0 if there is no such edge. The weights on the edges can be computed by minimizing the following objective function,

$$\min \sum_i \|\mathbf{x}_i - \sum_j W_{ij}\mathbf{x}_j\|^2$$

with constraints

$$\sum_j W_{ij} = 1, j = 1, 2, ..., m$$

Please see [11] for the details about how to solve the above minimization problem.

3. **Computing the Projections**: In this step, we compute the linear projections. Solve the following generalized eigenvector problem:

$$XMX^T\mathbf{a} = \lambda XX^T\mathbf{a} \tag{1}$$

where

$$X = (\mathbf{x}_1, \cdots, \mathbf{x}_m)$$
$$M = (I - W)^T(I - W)$$
$$I = diag(1, \cdots, 1)$$

It is easy to check that M is symmetric and semi-positive definite.

Let the column vectors $\mathbf{a}_0, \cdots, \mathbf{a}_{d-1}$ be the solutions of equation (1), ordered according to their eigenvalues, $\lambda_0 \leq \cdots \leq \lambda_{d-1}$. Thus, the embedding is as follows:

$$\mathbf{x}_i \to \mathbf{y}_i = A^T\mathbf{x}_i$$
$$A = (\mathbf{a}_0, \mathbf{a}_1, \cdots, \mathbf{a}_{d-1})$$

where $\mathbf{y}_i$ is a d-dimensional vector, and A is an $n \times d$ matrix.

4. Theoretical Justification

In this Section, we provide theoretical analysis of the NPE algorithm, which is fundamentally based on Locally Linear Embedding [11].

4.1. Optimal Linear Embedding

As we described previously, NPE is a linear approximation to Locally Linear Embedding. Different from Principal Component Analysis which preserves global structure, NPE preserves local manifold structure. Here, by ”local structure” we mean that each data point can be represented as a linear combination of its neighbors.

Recall that given a data set we first construct an adjacency graph on the data set. For each data point, we find its K nearest neighbors. In many cases, the data points might reside on a nonlinear submanifold, but it might be reasonable to assume that each local neighborhood is linear. Thus, we can characterize the local geometry of these patches by linear coefficients that reconstruct each data point from its neighbors. Reconstruction errors are measured by the cost function [11]:

$$\phi(W) = \sum_i \|\mathbf{x}_i - \sum_j W_{ij}\mathbf{x}_j\|^2$$

which adds up the squared distances between all the data points and their reconstructions. Note that, W_{ij} vanishes for distant data points. Consider the problem of mapping the original data points to a line so that each data point on the line can be represented as a linear combination of its neighbors with the coefficients W_{ij}. Let $\mathbf{y} = (y_1, y_2, \cdots, y_m)^T$ be such a map. A reasonable criterion for choosing a “good” map is to minimize the following cost function [11]

$$\Phi(\mathbf{y}) = \sum_i \left(y_i - \sum_j W_{ij}y_j\right)^2$$

under appropriate constraints. This cost function, like the previous one, is based on locally linear reconstruction errors, but here we fix the weights W_{ij} while optimizing the coordinates y_i.

Suppose the transformation is linear, that is, $\mathbf{y}^T = \mathbf{a}^T X$, where the i-th column vector of X is $\mathbf{x}_i$. We define

$$z_i = y_i - \sum_j W_{ij}y_j$$

which can be written in vector form,

$$\begin{aligned}\mathbf{z} &= \mathbf{y} - W\mathbf{y} \\ &= (I - W)\mathbf{y}\end{aligned}$$

Following some algebraic formulations, the cost function can be reduced to

$$\begin{aligned}\Phi(\mathbf{y}) &= \sum_i \left(y_i - \sum_j W_{ij}y_j\right)^2 \\ &= \sum_i (z_i)^2 \\ &= \mathbf{z}^T\mathbf{z} \\ &= \mathbf{y}^T(I - W)^T(I - W)\mathbf{y} \\ &= \mathbf{a}^T X(I - W)^T(I - W)X^T\mathbf{a} \\ &\doteq \mathbf{a}^T XMX^T\mathbf{a}\end{aligned}$$

where $M = (I-W)^T(I-W)$. Clearly, the matrix XMX^T is symmetric and semi-positive definite. In order to remove an arbitrary scaling factor in the projection, we impose a constraint as follows:

$$\mathbf{y}^T\mathbf{y} = 1 \Longrightarrow \mathbf{a}^T XX^T\mathbf{a} = 1$$

Finally, the minimization problem reduces to finding:

$$\underset{\substack{\mathbf{a} \\ \mathbf{a}^T XX^T\mathbf{a}=1}}{\arg\min} \;\; \mathbf{a}^T XMX^T\mathbf{a}$$

The transformation vector **a** that minimizes the objective function is given by the minimum eigenvalue solution to the following generalized eigenvector problem:

$$XMX^T\mathbf{a} = \lambda XX^T\mathbf{a}$$

It is easy to check that the matrices XMX^T and XX^T are symmetric and positive semi-definite. Sometimes the row vectors of X are linearly dependent, thus the matrix XX^T is singular. Suppose the rank of X is l. In this case, one can apply Singular Value Decomposition (SVD) to project the data points into a l-dimensional subspace in which the new data matrix $\widetilde{X}$ becomes non-singular:

$$X = USV^T$$

$$\widetilde{X} = U^T X = SV^T$$

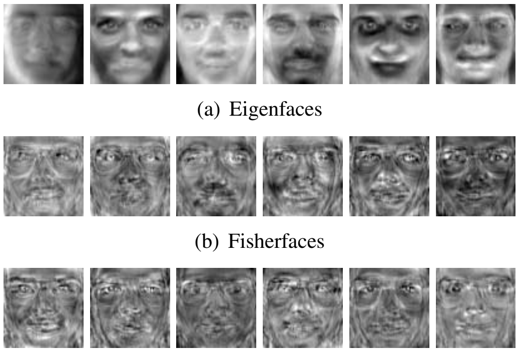
where $U = (\mathbf{u}_1, \cdots, \mathbf{u}_l)$, and $\mathbf{u}_i$ is the eigenvector of XX^T; $V = (\mathbf{v}_1, \cdots, \mathbf{v}_l)$, and $\mathbf{v}_i$ is the eigenvector of $X^T X$; and S is a $l \times l$ diagonal matrix whose entries are the non-zero singular values of X. Both S and V are of full rank, so $\widetilde{X}$ is also of full rank. In this way, the optimal projections are the eigenvectors of the matrix $(\widetilde{X}\widetilde{X}^T)^{-1}(\widetilde{X}M\widetilde{X}^T)$.

4.2. Connections to Locality Preserving Projection

Locality Preserving Projection (LPP) is a recently proposed linear dimensionality reduction algorithm [5]. In this Section, we discuss the connections between LPP and NPE.

LPP is a linear approximation to Laplacian Eigenmaps [2]. It is obtained by finding the optimal linear approximations to the eigenfunctions of the Laplace Beltrami operator on the manifold. LPP can be obtained by solving the following minimization problem,

$$\underset{\substack{\mathbf{a} \\ \mathbf{a}^T XDX^T\mathbf{a}=1}}{\arg\min} \;\; \mathbf{a}^T XLX^T\mathbf{a}$$



(a) Eigenfaces

(b) Fisherfaces

(c) NPEfaces

Figure 1: The first 6 basis vectors of Eigenfaces, Fisherfaces, and NPEfaces calculated from the face images in the ORL database.

where L is the so called *graph Laplacian* [3] induced from the graph structure. Specifically, $L = D - W$, where W is a pre-defined similarity matrix and D is a diagonal matrix, $D_{ii} = \sum_i W_{ij}$. As can be seen, LPP tries to minimize $\mathbf{a}^T XLX^T\mathbf{a}$, while NPE tries to minimize $\mathbf{a}^T XMX^T\mathbf{a}$. Note that, a vector $\mathbf{f} = (f_1, \cdots, f_m)$ can be thought of as a function defined on the graph such that f_i is the map of the i-th node. Thus, a matrix can be thought of as an operator acting on functions defined on the graph. In [2], Belkin and Niyogi show that under certain conditions

$$M\mathbf{f} \approx \frac{1}{2}L^2\mathbf{f}$$

Also, from spectral graph theory [3], we know that L provides a discrete approximation to the Laplace Beltrami operator $\mathcal{L}$ on the manifold. Therefore, the matrix M provides a discrete approximation to $\mathcal{L}^2$. This indicates that NPE essentially tries to find the linear approximations to the eigenfunctions of the iterated Laplacian $\mathcal{L}^2$. Eigenfunctions of $\mathcal{L}^2$ coincide with those of $\mathcal{L}$. In this sense, NPE and LPP provide two different ways to linearly approximate the eigenfunctions of the Laplace Beltrami operator.

5. Experimental Results

In this Section, we investigate the use of NPE on face analysis (representation and recognition). We compare our proposed algorithm with Eigenface [14] and Fisherface [1], two of the most popular linear techniques for appearance-based face recognition.

5.1. Face Representation using NPE

As we described previously, a face image can be represented as a point in image space. A typical image of size $m \times n$ describes a point in $(m \times n)$-dimensional image space. However, due to the unwanted variations resulting from changes in lighting, facial expression, and pose, the image space might not be an optimal space for visual representation and recognition.

1550-5499/05 $20.00 © 2005 IEEE

Figure 2: Sample face images from the ORL database. For each subject, there are 10 face images with different facial expression and details.

In Section 3, we have discussed how to learn a neighborhood preserving face subspace which is insensitive to outlier and noise. The images of faces in the training set are used to learn such a face subspace. The subspace is spanned by the basis vectors obtained from Eqn. (1). Therefore, any image in the face subspace can be represented as a linear combination of the basis vectors. We can display the basis vectors as a sort of feature images. Using the ORL face database as the training set, we present the first 6 basis vectors in Figure 1, together with Eigenfaces and Fisherfaces.

5.2. Face Recognition using NPE

PCA and LDA are the two most widely used subspace learning techniques for face recognition [1], [14]. These methods project the training sample faces to a low dimensional representation space where the recognition is carried out. The main supposition behind this procedure is that the face space has a lower dimension than the image space, and that the recognition of the faces can be performed in this reduced space. In this section, we investigate the use of NPE for face recognition.

5.2.1 Data Preparation

We use ORL face database in this work. In all the experiments, preprocessing to locate the faces was applied. Original images were normalized (in scale and orientation) such that the two eyes were aligned at the same position. Then, the facial areas were cropped into the final image for matching. The size of each cropped image in all the experiments is 32 × 32 pixels, with 256 gray levels per pixel. Thus, each image can be represented by a 1024-dimensional vector in image space. No further preprocessing is done. Different pattern classifiers have been applied for face recognition, including nearest-neighbor [14], Bayesian [9], and Support Vector Machines [10], etc. In this paper, we apply nearest-neighbor classifier for its simplicity.

5.2.2 Face Recognition on ORL Database

The ORL (Olivetti Research Laboratory) face database is used in this test. It consists of a total of 400 face images, of a total of 40 people (10 samples per person). The images were captured at different times and have different variations including expressions (open or closed eyes, smiling or non-smiling) and facial details (glasses or no glasses). The images were taken with a tolerance for some tilting and rotation of the face up to 20 degrees. 10 sample images of one individual are displayed in Figure 2. For each individual, $l (= 2, 3, 4, 5)$ images are randomly selected for training and the rest are used for testing.

For each given l, we average the results over 20 random splits. In general, the recognition rates varies with the dimension of the face subspace. Figure 3 shows the plots of error rate versus dimensionality reduction for the Eigenface, Fisherface, and NPE. For the baseline method, we simply performed face recognition in the original 1024-dimensional image space. The best result obtained in the optimal subspace and the corresponding dimensionality for each method are shown in Table 1. Note that, the upper bound of the dimensionality of Fisherface is $c - 1$ where c is the number of individuals.

As can be seen, our NPE algorithm performed the best for all the cases. The Fisherface method performed comparatively to NPE as the size of the training set increases. Moreover, the optimal dimensionality obtained by NPE and Fisherface is much lower than that obtained by Eigenface.

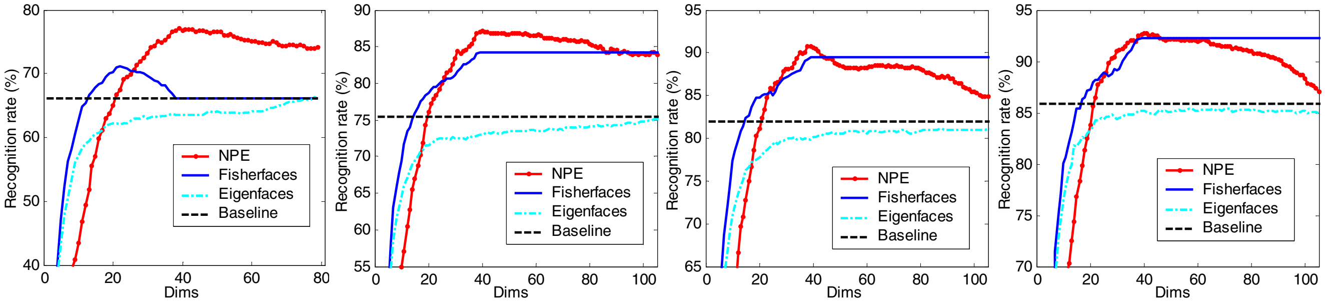
6. Conclusions

In this paper, we propose a novel linear dimensionality reduction algorithm called Neighborhood Preserving Embedding. It is a linear approximation to Locally Linear Embedding [11]. As a result it has similar neighborhood preserving properties. The main disadvantage of LLE is that, it is defined only on the training samples, and there is no natural maps of the testing sample. Instead, NPE is defined everywhere. Experiments on ORL face database have been conducted to demonstrate the effectiveness of our algorithm.

Several questions remain to be investigated in our future work,

1. As we described previously, both NPE and LPP [5] try to linearly approximate the eigenfunctions of the Laplace Beltrami operator $\mathcal{L}$ on the manifold. It is unclear how to evaluate these two methods in theory. More precisely, it is unclear how to define the optimal graph structure which can provide the best discrete approximation to $\mathcal{L}$.

2. When nearest neighbor search is involved, local structure seems to be more important than global structure. Thus, the algorithms preserving local structure, such as NPE and LPP, outperform the algorithms preserving global structure such as PCA and LDA. However, it remains unclear how to define the *locality* theoretically. Specifically, it remains unclear how to select the parameter k (or ϵ) in a principled manner.



(a) 2 Train (b) 3 Train (c) 4 Train (d) 5 Train

Figure 3: Recognition rate vs. dimensionality reduction on ORL database

Table 1: Performance comparisons on the ORL database

Method	2 Train	3 Train	4 Train	5 Train
Baseline	66.2%(1024)	75.4%(1024)	82.0%(1024)	85.9%(1024)
Eigenfaces	66.3%(78)	75.4%(119)	82.0%(159)	85.9%(199)
Fisherfaces	71.1%(22)	84.2%(39)	89.5%(39)	92.2%(39)
NPE	**77.1%(39)**	**87.1%(40)**	**90.8%(39)**	**92.7%(40)**

References

[1] P.N. Belhumeur, J.P. Hepanha, and D.J. Kriegman, “Eigenfaces vs. fisherfaces: recognition using class specific linear projection,”*IEEE. Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711-720, July 1997.

[2] M. Belkin and P. Niyogi, “Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering ,” *Advances in Neural Information Processing Systems 14*, Vancouver, British Columbia, Canada, 2001.

[3] Fan R. K. Chung, *Spectral Graph Theory,* Regional Conference Series in Mathematics, number 92, 1997.

[4] R. O. Duda, P. E. Hart, D. G. Stork, *Pattern Classification*, 2nd Edition, Wiley-Interscience, Hoboken, NJ, 2000.

[5] X. He and P. Niyogi, “Locality Preserving Projections ,” *Advances in Neural Information Processing Systems 16*, Vancouver, British Columbia, Canada, 2003.

[6] X. He, S. Yan, Y. Hu and H.-J. Zhang, “Learning a Locality Preserving Subspace for Visual Recognition ,” *Proc. 9th International Conference on Computer Vision*, Nice, France, 2003.

[7] K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman, “Video-Based Face Recognition Using Probabilistic Appearance Manifolds,” *IEEE Conference on Computer Vision and Pattern Recognition*, Madison, Wisconsin, 2003.

[8] Q. Liu, R. Huang, H. Lu, and S. Ma, “Face Recognition Using Kernel Based Fisher Discriminant Analysis,” *Fifth Int’l Conf. Automatic Face and Gesture Recognition*, 2002.

[9] B. Moghaddam, “Principal Manifolds and Probabilistic Subspaces for Visual Recognition,”*IEEE. Trans. Pattern Analysis and Machine Intelligence*, vol. 24, No. 6, June 2002.

[10] P. J. Phillips, “Support Vector Machines Applied to Face Recognition,”*Advance in Neural Information Processing Systems 11*, pp. 803-809, 1998.

[11] Sam Roweis, and Lawrence K. Saul, “Nonlinear Dimensionality Reduction by Locally Linear Embedding,” *Science*, vol 290, 22 December 2000.

[12] B. Schlkopf, A. Smola, and K.-R. Miller, “Nonlinear Component Analysis as a Kernel Eigenvalue Problem,” *Neural Computation*, vol. 10(5), 1998.

[13] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford, “A Global Geometric Framework for Nonlinear Dimensionality Reduction,” *Science*, vol 290, 22 December 2000.

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