Pattern Recognition 45 (2012) 1866–1876

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A two-dimensional Neighborhood Preserving Projection for appearance-based face recognition

Haijun Zhang [a,*], Q.M. Jonathan Wu [a], Tommy W.S. Chow [b], Mingbo Zhao [b]

[a] Department of Electrical and Computer Engineering, University of Windsor, 401 Sunset Avenue, Windsor, Ontario, Canada
[b] Department of Electronic Engineering, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong

ARTICLE INFO

Article history:
Received 25 March 2011
Received in revised form
1 November 2011
Accepted 3 November 2011
Available online 12 November 2011

Keywords:
Neighborhood Preserving Projection
Face recognition
Image analysis
Local method

ABSTRACT

This paper presents a two-dimensional Neighborhood Preserving Projection (2DNPP) for appearance-based face representation and recognition. 2DNPP enables us to directly use a feature input of 2D image matrices rather than 1D vectors. We use the same neighborhood weighting procedure that is involved in NPP to form the nearest neighbor affinity graph. Theoretical analysis of the connection between 2DNPP and other 2D methods is presented as well. We conduct extensive experimental verifications to evaluate the performance of 2DNPP on three face image datasets, i.e. ORL, UMIST, and AR face datasets. The results corroborate that 2DNPP outperforms the standard NPP approach across all experiments with respect to recognition rate and training time. 2DNPP delivers consistently promising results compared with other competing methods such as 2DLPP, 2DLDA, 2DPCA, ONPP, OLPP, LPP, LDA, and PCA.

1. Introduction

The last decade has witnessed growing interest in dimensionality reduction techniques for face recognition, in particular at the appearance-based aspect. Dimensionality reduction aims at mapping high-dimensional data to a lower dimensional space while preserving the intrinsic geometry of data samples by eliminating noises. In classification, as an application example, low-dimensional space usually advocates superior performance when low-dimensional data are used as inputs of a classifier. Amidst the rising popularity of dimensionality reduction techniques, appearance-based face recognition has made a huge progress. Widely used techniques include the following: Eigenfaces [1], a method employing Principal Component Analysis (PCA) on face images; Fisherfaces [2], a method employing Linear Discriminative Analysis (LDA) on face images; Laplacianfaces [3], a method employing Locality Preserving Projection (LPP) for face representation; Neighborhood Preserving Projection (NPP) [4] methods for face analysis and recognition.

PCA [1,5] is a well-known linear technique, which involves finding a set of mutually orthogonal basis functions and using the leading eigenvectors of the sample covariance matrix to characterize the lower dimensional space. Although PCA guarantees the ability to preserve the global structure, the locality of data samples is overlooked. This may lead to losing important information of the local geometry of each neighborhood. To incorporate discriminative information between classes into eigenspace, LDA [6], a supervised method for feature extraction and dimensionality reduction, has been widely used in many applications such as face recognition [2,7–10] and image retrieval [11]. But its projection is still based on the global structure of data samples. These observations have motivated the research in local methods (i.e. nonlinear manifold learning techniques) such as Isometirc Feature Mapping (Isomap) [12], Laplacian Eigenmap [13], and Locally Linear Embedding (LLE) [14]. But these techniques may yield *Out of Sample* problem, which can be described by the unclearness of how to evaluate the maps on new testing data. To overcome this shortcoming, corresponding projection algorithms are proposed. Deng et al. [15] developed an Isometric Projection algorithm by providing a functional mapping between the high and low dimensional spaces that are valid both on and off the training data. He et al. [3,16] designed an LPP algorithm and applied it to face recognition. This algorithm firstly constructs a weighted k-nearest neighbor (k-NN) graph to model the data topology. Then the projection in LPP is implemented by minimizing a certain objective function, which relates to a Laplacian matrix. The minimization leads to finding the transformation vector, which can be accomplished by solving a generalized eigenvalue problem. Gaussian weights are used in LPP attempting to amplify the neighborhood structure and to preserve it in the reduced space. But this strategy is somehow parameter sensitive, because the determination of weights requires the selection of an

* Corresponding author.
E-mail address: aarhzhang@gmail.com (H. Zhang).

0031-3203/$ - see front matter Crown Copyright © 2011 Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.11.002

appropriate value of the width of the Gaussian envelope [4]. Recently, Kokiopoulou and Saad [4] introduced several graph-based methods for dimensionality reduction. In particular, they proposed two algorithms, i.e. Orthogonal NPP (ONPP) and NPP, which are derived under the frame of LLE. These approaches use data-driven weights, which are found by solving a least-squares problem to reflect the intrinsic geometry of the local neighborhoods. The linear embedding in LLE is performed over the minimum of the reconstruction errors. This minimization problem leads to finding the d smallest eigenvalues (d is the dimension of the reduced space) and associated eigenvectors. The only difference between NPP and ONPP is that NPP imposes a condition of orthogonality on the projected data such that it needs to solve a generalized eigenvalue problem, while ONPP enforces the mapping to be orthogonal and solves an ordinary eigenvalue problem.

However, using the above-mentioned graph-based methods for face recognition usually involves a feature input of one-dimensional (1D) stacked vectors of face image matrices. Besides more computational time is required for training, this transformation from image matrix to stacked vector may have an impact on the evaluation of the covariance matrix such that the recognition rate is degraded. In view of these, Yang et al. [17] proposed a two-dimensional (2D) PCA algorithm, which directly uses image matrix as feature input. 2DPCA achieves promising results compared with the traditional PCA in terms of face recognition rate and training time. Motivated by 2DPCA, Li and Yuan [18] and Chen et al. [19] designed 2DLDA and 2DLPP algorithms, respectively. These two 2D extensions demonstrate promising properties compared with their 1D ones. In summary, we can say that these 2D methods have three advantages over 1D methods. First, the recognition rate is significantly improved since they accurately calculate the covariance matrix and somehow keep the relationships among sub-patterns in the reduced space. Second, they have effectively conquered the *Undersampled Size Problem*, which indicates that the number of samples is less than their dimension. Consequently, 2D approaches do not need any preprocessing procedures, whilst 1D methods usually use PCA to obtain initial projected space. Third, much less training time is required as 2D methods are based on direct matrix calculation that do not need initial PCA preprocess. In this paper, we design a 2DNPP algorithm, which is an extension of NPP, but it fully delivers the above-mentioned advantages of 2D methods. Likewise, it demonstrates superior property over other 2D methods such as 2DPCA [17], 2DLDA [18,30,31], and 2DLPP [19]. Concretely, 2DNPP has another two advantages. First, 2DNPP can be implemented in either an unsupervised or a supervised way, whilst 2DPCA [17] is only used in unsupervised cases and 2DLDA [18] is only conducted in supervised manner when the class labels are available. Second, 2DNPP does not require any parameter selection during the neighborhood weighting, whilst 2DLPP [19] is sensitive to the selection of the width of a Gaussian envelope [4]. To evaluate the performance of 2DNPP, this study has included extensive experimental verifications on different face datasets, i.e. ORL, UMIST, and AR datasets. The results show that 2DNPP achieves superior recognition rate compared with 2DPCA [17], 2DLPP [19] and 1D methods such as PCA [5], LDA [2], LPP [3], NPP [4], and ONPP [4]. Our proposed 2DNPP delivers comparable results compared with 2DLDA [18].

It is worthwhile to highlight the contributions of this research here: (1) we propose a 2DNPP algorithm, which is directly derived from the standard NPP algorithm; (2) we conduct detailed comparisons between 2DNPP and other competing algorithms, in particular the comparisons to other 2D techniques (i.e. 2DLPP, 2DLDA, and 2DPCA). As such, we are able to provide performance evaluations of these methods for other researchers; (3) in line with the essence of 2DNPP and 2DLPP, we also extend the ONPP and OLPP algorithms (we call the extensions, 2DONPP and 2DOLPP, respectively), and we evaluate them under our experimental framework. Interesting results have been shown that, unlike ONPP and OLPP, both 2DONPP and 2DOLPP failed in face recognition with very large error rate. This finding shows a big necessity of the constraint to the objective function $\Phi(Y)$ in the form of $YY^T=I_d$ (see Section 4.1) when it comes to dealing with two-dimensional (2D) feature inputs.

The remaining sections of this paper are organized as follows. A detailed description of 2DNPP algorithm in an unsupervised setting is presented in Section 2. In Section 3, we introduce a supervised version of 2DNPP algorithm. In Section 4, we conduct a detailed theoretical analysis of the connection between 2DNPP and other 2D methods. In Section 5, we summarize the face recognition procedures. In Section 6, we evaluate the performance of 2DNPP with respect to recognition rate and training time over three popular face datasets. Section 7 ends the paper with conclusion and future work propositions.

2. 2DNPP

In this section, we develop a 2DNPP algorithm to learn such a transformation matrix V that the locality property of the set X is preserved. 2DNPP is a two-dimensional extension of NPP [4], which is designed under the framework of LLE [14]. According to [14], given a well-sampled manifold, we expect each data point and its k nearest neighbors (k-NNs) to lie on or close to a locally linear patch of the manifold. Thus, we characterize the local geometry of these patches by linear coefficients that reconstruct each data point from its neighbors. In the following we introduce a method to learn the linear coefficients such that the reconstruction errors are minimized.

2.1. Problem statement

Given a set of N training face images $\{x_i\}$ ($x_i \in R^{m\times n}, i=1, 2, \ldots, N$), the goal of 2D dimensionality reduction is to design a linear transformation matrix V ($V \in R^{m\times d}$, where $d<m$) such that the set $\{x_i\}$ is projected to a lower dimensional space $\{y_i\}$, where $y_i \in R^{d\times n}$. Let $X=[x_1, x_2, \ldots, x_N]$ denote the image set $\{x_i\}$, and $Y=[y_1, y_2, \ldots, y_N]$ denote the projection set $\{y_i\}$, respectively. The linear transformation is given by

$$Y = V^T X. \tag{1}$$

Likewise, each image matrix x_i can be replaced by $y_i=V^T x_i$ directly. The projection Y is an accurate representation of the original data X, because the noises spread in the original space are eliminated and the desirable properties are preserved. Typical examples of desirable properties, which are similar to the case tackled by 1D methods [4], include the global geometry or locality information.

2.2. Neighborhood weighting

Consider an affinity matrix $W=[w_{ij}] \in R^{N\times N}$, in which each component of w_{ij} represents the optimal weight between the data point x_i and its neighbor x_j. According to the basic assumption that each data point and its k-NNs lie on a locally linear manifold, we can reconstruct each data point x_i with a linear combination of its k-NNs [4,14]. The reconstruction errors are measured by minimizing the following cost function:

$$\delta(W) = \sum_i \|x_i - \sum_j w_{ij} x_j\|_2^2, \tag{2}$$

where the weights w_{ij} are subject to the constraints as follows:

(1) $w_{ij}=0$, if x_j is not one of the k-NNs of x_i.
(2) $w_{ii} \equiv 0$ for all i.
(3) $\sum_j w_{ij} = 1$.

In addition, $\|\cdot\|_2$ is the Frobenius norm. The objective of the function is to find the optimal weights w_{ij}, which satisfy the above constraints. As a result, the reconstruction errors $\delta(W)$ are minimized. Let $s_i \in R^{mn \times 1}$ denote the stacked vector of x_i by column. It is easy to see that the format of the reconstruction errors illustrated in Eq. (2) is equivalent to the following representation:

$$\delta(W) = \sum_i \|s_i - \sum_j w_{ij} s_j\|_2^2. \tag{3}$$

Thus, the determination of the weights w_{ij} for 2DNPP is exactly the same with that conducted in NPP [4]. For completeness, we briefly summarize the process here. More details can be referred to [4,20].

Let $G \in R^{k \times k}$ denote the local Grammian matrix associated with the stacked vector s_i of data point x_i. The components of G is defined by

$$g_{uv} = (s_i - s_u)^T (s_i - s_v). \tag{4}$$

Let $S^{(i)}$ be a system of stacked vectors with respect to x_i and its neighbors, and to find the optimal weights $w_{i,:}$, we need to solve the least squares $(S^{(i)} - s_i e^T) w_{i,:} = 0$ subject to the constraint $e^T w_{i,:} = 1$, where $e = [1, \ldots, 1]^T$ is the vector of all ones. The solution of this constrained least squares problem is given by the following, which involves the inverse of G [4,14,20]:

$$w_{i,:} = \frac{G^{-1} e}{e^T G^{-1} e}. \tag{5}$$

This is an explicit expression for the calculation of the weights. As pointed out in [4], determining the weights $w_{i,j}$ for a given data point x_i is a local calculation, because it only involves x_i and its neighbors. Any algorithm for this computation will be fairly inexpensive. It is noted that the Grammian matrix G may be a singular matrix. In practice, we can set $G' = G + \sigma I_k$ to replace G as the Grammian matrix. Here, σ is a small number (usually set to 10^{-4}), and I_k is the identify matrix of order k. This setting is similar to that in [20].

2.3. Projection learning

Given that each image matrix $x_i \in R^{m \times n}$ is projected to a matrix with lower size $y_i \in R^{d \times n}$, $d < m$. Similar to LLE [14] and NPP [4], we use the same weights, which are used for reconstructing the data point x_i by its neighbors in the original space, to reconstruct its projection y_i in the lower dimensional space by its corresponding neighbors. The determination of y_i is given by minimizing the following cost function [14]:

$$\Phi(Y) = \sum_i \|y_i - \sum_j w_{ij} y_j\|_2^2. \tag{6}$$

Since the weights $W = [w_{ij}] \in R^{N \times N}$ are fixed, the goal of Eq. (6) is to minimize the objective $\Phi(Y)$ associated with $Y = [y_1, y_2, \ldots, y_N] \in R^{d \times nN}$.

Note that we can rewrite the objective $\Phi(Y)$ in the following by employing some simple algebraic steps:

$$\begin{aligned}\sum_i \|y_i - \sum_j w_{ij} y_j\|_2^2 &= \|Y - Y(W \otimes I_n)^T\|_2^2 = \|Y(I \otimes I_n - W^T \otimes I_n)\|_2^2 \\ &= tr\{Y[(I-W^T) \otimes I_n][(I-W) \otimes I_n]Y^T\} = tr\{Y[(I-W^T)(I-W) \otimes I_n]Y^T\} \\ &= tr\{V^T X[(I-W^T)(I-W) \otimes I_n]X^T V\},\end{aligned} \tag{7}$$

where operator $\otimes$ is the Kronecker product of the matrices, I_n is the identity matrix of order n, and I is the identity matrix of order N. Here, we impose a constraint to the objective $\Phi(Y)$ by $YY^T = I_d$, where I_d is the identity matrix of order d. The goal of imposing this constraint is to produce an orthogonal projection set Y. Thus, we have the complete format of the constraint as follows:

$$V^T X X^T V = I_d. \tag{8}$$

Finally, the minimization problem is summarized as

$$\underset{V}{\text{argmin}} \quad tr\{V^T X[(I-W^T)(I-W) \otimes I_n]X^T V\}$$
$$\text{subject to} V^T X X^T V = I_d. \tag{9}$$

It is worth noting that the matrices: $X[(I-W^T)(I-W) \otimes I_n]X^T$ and $XX^T = XIX^T$ are both symmetric and positive semidefinite since the matrix $[(I-W^T)(I-W) \otimes I_n]$ and the identity matrix I are both symmetric and positive semidefinite. Thus, similar to LPP [3], the above minimization problem leads to solving the following generalized eigenvalue problem, which is derived from spectral graph theory [13,21]:

$$X[(I-W^T)(I-W) \otimes I_n]X^T V = \lambda X X^T V, \tag{10}$$

where λ is the eigenvalue solution to the problem. The solution V to the optimization problem is the basis of the eigenvectors associated with the d smallest eigenvalues for the generalized eigenvalue problem as shown in Eq. (10).

3. Supervised 2DNPP

In the last section, we described an unsupervised version of 2DNPP algorithm without considering label information of training faces. It may require prior knowledge to select the value of k, which is the number of NNs associated with each training sample x_i. Similar to the supervised NPP [4] algorithm, it is easy to extend 2DNPP to a supervised setting when the labels of training faces are available. In a supervised setting, the NNs of a data sample x_i is only considered if they belong to the same class with x_i. By this simple setting, we incorporate the discriminative information into the neighborhood weighting of 2DNPP. Thus, 2DNPP is able to deliver not only intrinsic geometry information but also discriminative information, which enables us to further improve the recognition performance.

3In a supervised setting, let c denote the number of classes with respect to training samples, and let N_i be the number of data points that belong to the ith class. Recall that N represents the total number of training samples, hence, $N = \sum_{i=1}^{c} N_i$. As mentioned in Section 2.2, the weight matrix W involved in 2DNPP can be achieved by following the same procedures used for NPP [4]. Therefore, the weight matrix W inherits the same property described in NPP [4]. For instance, the rank of $(I-W)$ is at most $(N-c)$. The detailed proof can be found in [4]. Consequently, the matrix $\widehat{M} = X(I-W^T)(I-W)X^T$ will have rank at most $(N-c)$. This will directly lead to the *Undersampled Size Problem*. In order to avoid the singularity of the matrix $\widehat{M}$, an initial PCA [5] projection is usually employed such that the dimensionality of the data vectors is reduced to $(N-c)$ [4]. For the case of 2DNPP, we can easily conclude that the rank of $(I-W) \otimes I_n$ is at most $n(N-c)$.

However, the resulting matrix $\tilde{M}=X[(I-W^T)(I-W)\otimes I_n]X^T \in R^{m\times m}$ will not be singular anymore, because usually $m \ll n(N-c)$ in practice. Thus, 2DNPP does not require the PCA preprocessing. In this sense, it saves significant training time compared with the standard NPP [4].

4. Relationship analysis

In this section, we conduct a deep analysis of 2DNPP to build a relationship with 2DLPP [19], 2DPCA [17], and 2DLDA [18].

4.1. Relationship to 2DLPP

Similar to 2DNPP, 2DLPP [19] seeks to preserve the locality property of data samples in the reduced space but minimizes the following cost function:

$$\Phi_{2DLPP}=\frac{1}{2}\sum_{i,j=1}^{N} S_{ij}\|x_i-x_j\|_2^2, \tag{11}$$

where S_{ij} denotes the weight between data samples x_i and x_j. The weight matrix S, which is symmetric, can be defined by either a radius setting of the local neighborhood or a Heat Kernel. For details, refer to [3,19]. By following some algebraic steps, the objective function can be represented by

$$\Phi_{2DLPP}=tr\{V^TX[(D-S)\otimes I_n]X^TV\}, \tag{12}$$

where $D=\mathrm{diag}(d_i)$ with $d_i=\sum_{j=1}^{N}S_{ij}$, $L=(D-S)$ is the Laplacian matrix [21]. By imposing a constraint $V^TX(D\otimes I_n)X^TV=I_d$, 2DLPP leads to the following minimization problem:

$$\arg\min_V \quad tr\{V^TX[(D-S)\otimes I_n]X^TV\}$$
$$\text{subject to } V^TX(D\otimes I_n)X^TV=I_d. \tag{13}$$

However, if we first normalize the weight matrix S by the setting that each entry is divided by its corresponding row sum, the diagonal matrix D will become an identity matrix I of order N, and the Laplacian matrix $L=(I-\tilde{S})$, where $\tilde{S}$ is the normalized version of S. Then the minimization problem involved in 2DLPP will become

$$\arg\min_V tr\{V^TX[(I-\tilde{S})\otimes I_n]X^TV\}$$
$$\text{subject to } V^TXX^TV=I_d. \tag{14}$$

The solution V to above problem is to solve the generalized eigenvalue problem and to find the basis of the eigenvectors associated with the d smallest eigenvalues. By comparing with Eq. (9), we can easily observe that the only difference between 2DLPP and 2DNPP is the matrix including the weights between data samples. 2DNPP uses the matrix $M_{2DNPP}=(I-W^T)(I-W)$ to evaluate the intrinsic geometry of local neighborhoods, whilst 2DLPP employs $M_{2DLPP}=(I-\tilde{S})$. Here, M_{2DLPP} acts like a counterpart to M_{2DNPP} but is defined by an artificial setting to the weight matrix. In contrast, 2DNPP assumes that each data sample, along with its k-NNs lies on a locally linear manifold [4]. Thus, given the value of k, the number of NNs associated with each training sample, the implementation of the weight matrix will be straightforward. In addition, 2DNPP is equivalent to 2DLPP in some case, as described by the following proposition.

Proposition 1. *When we put 1/N at each entry of the weight matrices W and S, i.e.* $w_{ij}=1/N$ *and* $S_{ij}=1/N$*, 2DNPP is equivalent to 2DLPP.*

Proof. If $S_{ij}=1/N$ involved in 2DLPP, the diagonal matrix D becomes an identity matrix I of order N, and the Laplacian matrix $L=(D-S)$ is written as $L=(I-(1/N)ee^T)$, where $e=[1,\ldots,1]^T$ is a column vector with the length N. On the other hand, if $w_{ij}=1/N$ used in 2DNPP, the weight matrix W becomes a symmetric matrix, and the matrix $M_{2DNPP}=(I-W^T)(I-W)$ can be rewritten as

$$M_{2DNPP}=(I-W)^2=\left(I-\frac{1}{N}ee^T\right)^2=I-\frac{2}{N}ee^T+\frac{1}{N^2}e(e^Te)e^T$$
$$=I-\frac{2}{N}ee^T+\frac{N}{N^2}ee^T=I-\frac{1}{N}ee^T, \tag{15}$$

which is exactly same with the Laplacian matrix L. Therefore, when taking $1/N$ at each entry of the weight matrices W and S, i.e. $w_{ij}=1/N$ and $S_{ij}=1/N$, 2DNPP is equivalent to 2DLPP. It is worth noting that this proposition is also satisfied for that of NPP and LPP in 1D case. □

Table 1
Problems involved in different methods.

Method	Minimized objective function	Constraint	Eigenvalue problem
2DNPP	$tr\{V^TX[(I-W^T)(I-W)\otimes I_n]X^TV\}$	$V^TXX^TV=I_d$	Generalized
NPP	$tr\{V^TX[(I-W^T)(I-W)]X^TV\}$	$V^TXX^TV=I_d$	Generalized
2DONPP	$tr\{V^TX[(I-W^T)(I-W)\otimes I_n]X^TV\}$	–	Ordinary
ONPP	$tr\{V^TX[(I-W^T)(I-W)]X^TV\}$	–	Ordinary
2DLPP	$tr\{V^TX[(D-S)\otimes I_n]X^TV\}$	$V^TX(D\otimes I_n)X^TV=I_d$	Generalized
LPP	$tr\{V^TX(D-S)X^TV\}$	$V^TXDX^TV=I_d$	Generalized
2DOLPP	$tr\{V^TX[(D-S)\otimes I_n]X^TV\}$	–	Ordinary
OLPP	$tr\{V^TX(D-S)X^TV\}$	–	Ordinary

Note: For 2D methods, feature input $X\in R^{m\times nN}$; for 1D methods, feature input $X\in R^{m\times N}$.

As discussed above, NPP and LPP are equivalent under some special setting. It is worth pointing out that the only difference between NPP, LPP and their orthogonal versions (i.e. ONPP and OLPP [4]) is that NPP and LPP impose a constraint to the objective function $\Phi(Y)$ in the form of $YY^T=I_d$ such that they need to solve a generalized eigenvalue problem, while ONPP and OLPP enforce the mapping V to be orthogonal and solves an ordinary eigenvalue problem. The imposition of this orthogonality does boost the performance of NPP and LPP in a significant rate [4]. In line with the essence of this in 1D case, we can easily make extensions to ONPP and OLPP for 2D feature inputs. We call the extension algorithm for ONPP, 2DONPP, which does not consider the constraint as shown in Eq. (8). Likewise, we call the extension algorithm for OLPP, 2DOLPP, which optimizes the objective function without taking the constraint into account as shown in Eq. (13). For clarity, we summarize the problems involved in different methods in Table 1. With these extensions, interesting results have been shown in our experiments. Basically, we found that both 2DONPP and 2DOLPP failed in face recognition with very large error rates (see Section 6).

4.2. Relationship to 2DPCA

2DPCA [17] seeks the transformation matrix V by maximizing the covariance of the projected image matrices. Let us put $1/N$ at each entry of the weight matrix W, which is used in 2DNPP, i.e. $w_{ij}=1/N$. Thus, W becomes a symmetric matrix. According to proposition 1, the matrix $M_{2DNPP}=(I-W^T)(I-W)$ can be represented as $M_{2DNPP}=I-(1/N)ee^T$. We have

$$\frac{1}{N}\tilde{M}=\frac{1}{N}X\left[(I-W^T)(I-W)\otimes I_n\right]X^T=X\left[\left(\frac{1}{N}I-\frac{1}{N^2}ee^T\right)\otimes I_n\right]X^T$$
$$=\sum_{i=1}^{N}x_i\left(\frac{1}{N}I_n\right)x_i^T-\sum_{i,j=1}^{N}x_i\left(\frac{1}{N^2}I_n\right)x_j^T=\frac{1}{N}\sum_{i=1}^{N}(x_i-\bar{x})(x_i-\bar{x})^T=\tilde{M}_{2DPCA}, \tag{16}$$

where $\bar{x}=\sum_{i=1}^{N} x_i/N$ is the mean of data samples, and $\tilde{M}_{2DPCA}$ is just the covariance matrix of data samples. However, 2DPCA aims at finding the eigenvectors of the matrix $\tilde{M}$ associated with the largest eigenvalues such that the covariance of data samples is maximized [3,19]. In contrast, 2DNPP aims at preserving the local geometry of data samples and chooses the basis of the matrix $\tilde{M}$ associated with the smallest eigenvalues. The rationale behind this scheme is that the directions preserving the locality property are those minimizing the local covariance rather than minimizing the global covariance. The detailed explanation can be found in [3].

4.3. Relationship to 2DLDA

2DLDA [18] seeks a projection that carries maximal discriminative information. This projection is accomplished by solving the following generalized eigenvalue problem:

$$S_b V = \lambda S_w V, \tag{17}$$

where S_b and S_w are called the within-class scatter matrix and the between-class scatter matrix, respectively, and they are defined as follows:

$$S_b = \sum_{k=1}^{q} N_k(\bar{x}^{(k)}-\bar{x})(\bar{x}^{(k)}-\bar{x})^T, \tag{18}$$

$$S_w = \sum_{k=1}^{q}\sum_{j=1}^{N_k}(x_j^{(k)}-\bar{x}^{(k)})(x_j^{(k)}-\bar{x}^{(k)})^T, \tag{19}$$

where $\bar{x}^{(k)}$ is the mean of data samples that belong to the kth class, $x_j^{(k)}$ is the jth data sample in the kth class, N_k is the number of data samples in the kth class, and q is the number of classes in the data set. If we set $w_{ij}=1/N_k$ for the weight between data samples $x_i^{(k)}$ and $x_j^{(k)}$, otherwise $w_{ij}=0$, S_w, and S_b can be rewritten as

$$\begin{aligned} S_w &= \sum_{k=1}^{q}\sum_{j=1}^{N_k}(x_j^{(k)}-\bar{x}^{(k)})(x_j^{(k)}-\bar{x}^{(k)})^T \\ &= \sum_{k=1}^{q}\sum_{j=1}^{N_k} x_j^{(k)}(x_j^{(k)})^T - \sum_{k=1}^{q} N_k\bar{x}^{(k)}(\bar{x}^{(k)})^T \\ &= \sum_{k=1}^{q} x^{(k)}\left[\left(I_{N_k}-\frac{1}{N_k}e_{N_k}e_{N_k}^T\right)\otimes I_n\right](x^{(k)})^T \\ &= \sum_{k=1}^{q} x^{(k)}[(I_{N_k}-W^{(k)})^2\otimes I_n](x^{(k)})^T \\ &= X[(I-W^T)(I-W)\otimes I_n]X^T, \end{aligned} \tag{20}$$

$$\begin{aligned} S_b &= \sum_{k=1}^{q} N_k(\bar{x}^{(k)}-\bar{x})(\bar{x}^{(k)}-\bar{x})^T \\ &= \sum_{k=1}^{q} N_k\bar{x}^{(k)}(\bar{x}^{(k)})^T - N\bar{x}(\bar{x})^T \\ &= X\left[(I-\frac{1}{N}ee^T)\otimes I_n\right]X^T - X\left[(I-W)^2\otimes I_n\right]X^T \\ &= N\tilde{M}_{2DPCA} - X[(I-W^T)(I-W)\otimes I_n]X^T. \end{aligned} \tag{21}$$

Thus, it leads to the following generalized eigenvalue problem:

$$X\left[(I-W^T)(I-W)\otimes I_n\right]X^T V = \frac{N}{1+\lambda}\tilde{M}_{2DPCA}V. \tag{22}$$

Then we choose the eigenvectors associated with the smallest eigenvalues as the optimal projection. If we set the mean of data samples to zero, the matrix $N\tilde{M}_{2DPCA}$ becomes XX^T. Thus, Eq. (22) is very similar to Eq. (10) defined in Section 2.3. Note that the above derivation of the relationship between 2DNPP and 2DLDA is identical with that of the connection between 2DLLP and 2DLDA [19]. For more details, refer to [3,19]. Therefore, we see that 2DNPP, 2DLPP, and 2DLDA are much related, and they can be unified in a single framework under certain condition.

5. Face recognition

For clarity, in this section we summarize the 2DNPP algorithm and its application procedures on face recognition.

First, we input the training samples and train a 2DNPP algorithm. After learning the transformation matrix V, we are able to directly map the training data to the lower dimensional space. Likewise, given a new testing image matrix x_t, it is quite trivial to project it to the image y_t located in the subspace by simple matrix product as follows:

$$y_t = V^T x_t. \tag{23}$$

Consequently, the distance metric between two face samples can be defined and measured onto the subspace. In this paper, the distance metric is defined by the following [17]:

$$dis(y_i,y_j) = \sum_{l=1}^{d} ||y_i^{(l)}-y_j^{(l)}||_2, \tag{24}$$

where $y_i \in R^{d\times n}$ represents the ith feature matrix, and $y_i^{(l)} \in R^{1\times n}$ is the lth row of y_i. The smaller the distance between two feature matrices is, the closer the two faces are. Then a classifier can be used for face recognition phase. Here, we use NN classifier [2] to perform recognition in low dimensional space. Other pattern classifiers such as Support Vector Machine [22], Bayesian [23], and neural networks [24] can also be employed.

The overall face recognition procedure is summarized as follows:

(1) Input the training set $X=(x_1, x_2, \ldots, x_N)\in R^{m\times nN}$ and the dimension of reduced space d.
(2) Compute the k nearest neighbors of each data sample x_i.
(3) Compute the weights $w_{i.}$ associated with data sample x_i and its neighbors using Eq. (5), and construct the weight matrix W.
(4) Solve the generalized eigenvalue problem as shown in Eq. (10), and construct the mapping V whose column vectors are taken from the eigenvectors associated with the d smallest eigenvalues.
(5) Compute the projected training images $y_i=V^T x_i$.
(6) Given a testing image x_t, map it onto the subspace by $y_t=V^T x_t$.
(7) Input the projected testing image together with the projected training images to a NN classifier, and classify the testing image to corresponding class learned by the classifier.

6. Experiments

In this section, we evaluate the performance of 2DNPP algorithm on three popular datasets, i.e. ORL [25], UMIST [26], and AR [27] face datasets. First, we compare the recognition results of 2DNPP with those of other techniques including 2DLPP [19], 2DONPP, ONPP [4], NPP [4], 2DOLPP, OLPP [4], and LPP [3] (see Section 6.1). For completeness, we also include the comparative results of discriminative information based methods (i.e. 2DLDA [18] and LDA [2]) and global geometry preserving based methods (i.e. 2DPCA [17] and PCA [1]). Second, we test the performance of 2DNPP with respect to training time and demonstrate the comparative results with NPP [4] (see Section 6.2). Third, we investigate the effects of dimension variation of the reduced space for 2D methods (see Section 6.3). Here, due to the *Undersampled Size Problem*, ONPP [4], NPP [4], OLPP [4], LPP [3], and LDA [2] employs an initial PCA [5] projection to reduce the dimensionality of the data vectors to $(N-c)$ [4]. For simplicity, all the graph-based methods

Fig. 1. Image sample from ORL dataset: 10 different faces for each subject with variations in facial expressions, facial details, and poses.

including 2DNPP, 2DLPP [19], 2DONPP, ONPP [4], NPP [4], 2DOLPP, OLPP [4], and LPP [3] are based on a supervised setting. 2DLPP [19], 2DOLPP, OLPP [4], and LPP [3] use Gaussian weights. As suggested by [4], we determine the value of the width τ of the Gaussian envelop as follows: First, sample 1,000 points randomly and then calculate the pairwise distances among them. The width τ is set to half the median of those pairwise distances [4]. In addition, it is worth noting that the rank of the between-class scatter matrix in LDA [2] is at most $(c-1)$. Thus, the dimension of reduced space using LDA [2] is at most $(c-1)$.

6.1. *Performance of recognition rate*

6.1.1. *ORL*

The ORL dataset [25] contains 40 individuals. Each of them includes 10 different images, which show variations in facial expressions (smiling or not smiling), facial details (glasses or no glasses), and poses. All images are grayscale, and the original size of each image is 112×92. We resized the images to 64×64 for computational efficiency. Fig. 1 shows one sample subject of the ORL dataset. The dataset was firstly split into a training set and a testing set. 200 training images were randomly selected from the 40 subjects, i.e. 5×40. The remaining 200 images were used for evaluating the performance of recognition algorithms. 2D methods perform with the size of the reduced space $d \times 64$, where d is the number of selected eigenvectors and varies from 1 to 20 at the increment of 1. We test 1D methods with the dimension d_{1D} of the reduced 1D vectors from 5 to 150 at the increment of 5. For each value of d and d_{1D}, we calculate the average error rate across 20 different random realizations of the training/testing set. For both 2D methods and 1D methods, we then select the number of eigenvectors, d, which delivers the best performance with respect to the average error rate. Many researchers have used this setting in their experiments [4,17,19]. We also include the study of the effect of dimension selection on recognition rate (see Section 6.3). The comparative results are shown in Table 2.

From Table 2, we can observe that 2DNPP outperforms NPP and ONPP significantly. It is interesting to notice that OLPP delivers better results compared with 2DNPP and 2DLPP. But OLPP requires much more training time than 2DNPP and 2DLPP (see Section 6.2). It is surprising to find that 2DLDA, 2DPCA, LDA, and PCA all exhibit very competitive results. This suggests that preserving either the global geometry of samples or the discriminative information among samples is effective to raise the recognition performance. Preservation of the local geometry may cause the degradation of recognition rate for ORL dataset. In addition, we note that 2DONPP and 2DOLPP show the worst performance. It indicates that without considering

Table 2
Comparative results of different methods on ORL face dataset.

Method	d	Error rate (%)
2DNPP	9	5.95
2DONPP	20	85.58
2DLPP	9	6.07
2DOLPP	5	87.02
NPP	115	16.92
ONPP	150	9.30
LPP	25	17.92
OLPP	125	4.55
2DLDA	9	4.15
2DPCA	12	4.45
LDA	39	7.75
PCA	150	5.75

the constraint to the objective function involved in NPP and LPP degrades the overall performance significantly. In general, 2D methods expect for 2DONPP and 2DOLPP bring larger gain of recognition rate than 1D methods. It implies that 2D methods are capable of characterizing more accurate covariance of data samples. Besides, 2D methods bring large time efficiency gain as well (see Section 6.2).

6.1.2. *UMIST*

The UMIST dataset [26] contains 20 individuals. The number of images in each subject varies from 19 to 48 under different poses. The entire dataset consists of 575 face images. We used a cropped version of this dataset, which is online available.[1] All images are grayscale, and the original size of each image is 112×92. Likewise, we resized the images to 64×64. Fig. 2 shows an example of one subject of the UMIST dataset under 20 poses. We first split the whole dataset into a training set and a testing set. 10 images were randomly selected from each subject to construct the training set. Thus the total number of training images is 200. The remaining 375 images were used for testing. 2D methods perform with the size of the reduced space $d \times 64$, where d is the number of selected eigenvectors and varies from 1 to 20 at the increment of 1. 1D methods are tested with the dimension d_{1D} of the reduced 1D vectors from 5 to 150 at the increment of 5. For each value of d and d_{1D}, we calculate the average error rate across 20 different random realizations. For both 2D methods and 1D methods, we then select the dimension of the reduced space, which delivers the best performance with respect to the average error rate. The comparative results are summarized in Table 3.

[1] http://www.cs.toronto.edu/~roweis/data.html.

Fig. 2. Image sample from UMIST dataset under different poses.

It is clear to see that 2DNPP delivers lower error rate than other graph-based methods such as 2DLPP, NPP, ONPP, LPP, and OLPP. In particular, 2DNPP achieves improved recognition rate by at least 4% compared with NPP and ONPP. In this testing, 2DLDA performs slightly better than 2DNPP. We observe that 2DLPP and 2DLDA characterize smaller dimension of the reduced space, i.e. 3×64, to deliver the best performance. It is also interesting to see that 1D PCA performs even better than 2DPCA in an insignificant rate in this case. Again, 2DONPP and 2DOLPP show big error rates.

6.1.3. AR

The AR dataset [27] contains over 4000 color images of 126 individuals, which include 70 men and 56 women. The images of each subject were taken in two sessions. In each session, each individual consists of 13 images, which show variations of frontal views of faces in facial expressions, lighting conditions, and occlusions. In this research, we took 100 individuals (50 men and 50 women) to experiment and used the first 13 images of each person to test the performance of all the algorithms. Thus, the total number of images used in this experiment is 1300, i.e. 13×100. We conducted tests on a cropped and grayscale version of these images and resized them to 64×64. Fig. 3 illustrates the color samples of one subject, where Fig. 3(a)–(d) shows the variations in facial expressions, Fig. 3(e), (f), (g), (i), (j), (l), and (m) shows the variations under different lighting conditions with/without occlusions, and Fig. 3(h)–(m) shows the variations in occlusions with/without lighting conditions. To fully evaluate the performance of 2DNPP together with other techniques, in the following, we make three tests based on variations in facial expressions, lighting conditions, and occlusions.

6.1.3.1. Facial expressions. In this test, we used images shown in Fig. 3(a)–(d) from each subject to evaluate the performance of all the algorithms handling different facial expressions. We randomly selected two images from Fig. 3(a)–(d) for training. Another two images are used for testing. Thus, the total number of training samples is 200, i.e. 2×100. The comparative results are shown in Table 4, where the average error rate of each method is based on 20 different random realizations. We observe that 2DNPP achieves better results than other graph-based methods, i.e. 2DLPP, ONPP, NPP, LPP, and OLPP. In particular, 2DNPP delivers over 10% of recognition rate compared with NPP and LPP. 2DNPP is more effective than 2DPCA, LDA, and PCA although 2DLDA performs slightly better than 2DNPP.

6.1.3.2. Lighting conditions. This experiment is to test 2DNPP together with other methods under varying lighting conditions. For training, we selected images shown in Fig. 3(a), (h), and (k) from each subject. For testing, we considered images shown in Fig. 3(e), (f), (g), (i), (j), (l), and (m). Thus, the total number of training samples is 300, i.e. 3×100, whilst the testing set includes 700 images, i.e. 7×100. We summarized the results in Table 5.

Table 3
Comparative results of different methods on UMIST face dataset.

Method	d	Error rate (%)
2DNPP	9	5.03
2DONPP	5	69.51
2DLPP	3	6.01
2DOLPP	5	73.89
NPP	135	14.83
ONPP	150	9.23
LPP	20	13.27
OLPP	45	5.43
2DLDA	3	3.80
2DPCA	13	6.97
LDA	19	7.08
PCA	125	6.68

It is clear to see that 2DNPP performs much better than other techniques. 2DNPP achieves over 5% of performance improvement compared with 2DLPP and ONPP and delivers over 10% of improvement of recognition rate compared with NPP, LPP, and OLPP. 2DPCA and PCA perform the worst due to considering only global geometry of data samples. It is interesting to observe that 1D LDA delivers much better results than 2DLDA, which performs the best in previous experiments.

6.1.3.3. Occlusions. Coping with occlusions is the most difficult task for face recognition, in particular with the use of dimensionality reduction methods. Here, we also include the testing results under different occlusions. The training images were taken from the samples illustrated in Fig. 3(a)–(f) from each person. The total number of training samples is 600. On the other hand, the testing images were the samples shown in Fig. 3(h)–(m) from each subject. Thus, 600 images construct the testing set. Table 6 shows the comparative results of all the algorithms. Again, 2DNPP delivers the best recognition rate among all the algorithms. 2DLDA performs slightly worse than 2DNPP. 2DLPP, OLPP, and LDA also achieve promising results. But LPP almost fails to recognize and delivers the worst performance. It is worth noting that 2DNPP improves the recognition rate by around 33% compared with NPP and ONPP. From the preceding experiments on the AR dataset, we observe that 2DONPP and 2DOLPP achieve only around 30% recognition rate.

6.2. Performance of training time

2D methods are based on direct matrix calculation and do not require initial PCA preprocess, whilst 1D methods such as NPP, ONPP, LPP, OLPP, and LDA need transform image matrices to flat vectors and must employ PCA preprocessing due to the *Undersampled Size Problem*. On the other hand, usually 2D methods require finding only a smaller number of eigenvectors. In this sense, 2D methods will save a large amount of time for training. In this sub-section, we experimentally evaluate the time

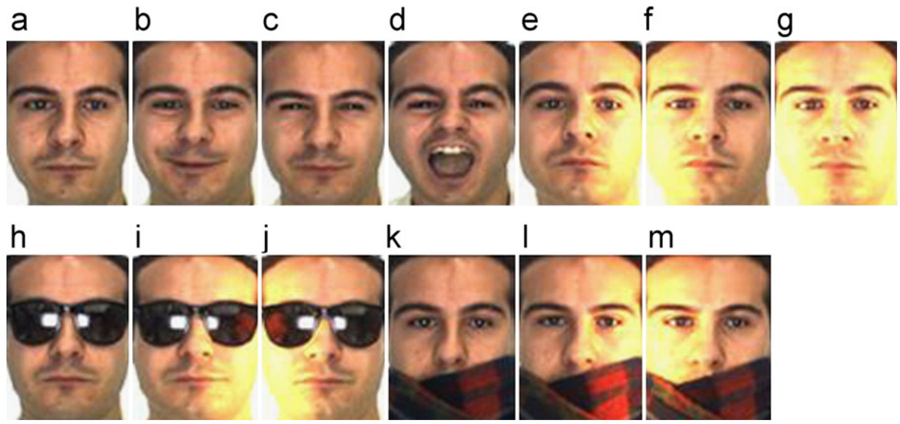

Fig. 3. Color image sample from AR dataset under variations in facial expressions, lighting conditions, and occlusions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4
Comparative results of different methods on AR dataset under varying facial expressions.

Method	*d*	Error rate (%)
2DNPP	5	3.68
2DONPP	14	72.08
2DLPP	5	6.77
2DOLPP	14	72.08
NPP	30	15.02
ONPP	70	3.73
LPP	30	19.93
OLPP	50	9.00
2DLDA	6	1.20
2DPCA	10	8.22
LDA	35	11.48
PCA	95	15.50

Table 5
Comparative results of different methods on AR dataset under varying lighting conditions.

Method	*d*	Error rate (%)
2DNPP	11	4.86
2DONPP	7	60.86
2DLPP	7	11.57
2DOLPP	7	61.57
NPP	125	16.57
ONPP	130	10.86
LPP	80	21.14
OLPP	60	32.43
2DLDA	20	27.43
2DPCA	20	49.43
LDA	90	8.57
PCA	175	77.71

performance of 2DNPP. Note that the training time of other 2D algorithms including 2DLPP, 2DLDA, and 2DPCA is similar to that of 2DNPP, and NPP requires the similar training time compared with other 1D methods such as ONPP, LPP, OLPP, LDA, and PCA. Therefore, we only compare 2DNPP with NPP in terms of training time in this experiment. The results are summarized in Table 7. We observed that the training time of 2DNPP is less than 1 s across all the experiments. The training of 2DNPP is over 400 times faster than that of NPP.

6.3. Study of dimension selection

In this experiment, we study the effect of varying the number of selected eigenvectors for 2D methods, which include 2DNPP,

Table 6
Comparative results of different methods on AR dataset under varying occlusions.

Method	*d*	Error rate (%)
2DNPP	16	42.17
2DONPP	8	75.33
2DLPP	12	47.17
2DOLPP	8	75.67
NPP	65	75.33
ONPP	75	75.67
LPP	200	89.50
OLPP	180	52.67
2DLDA	17	43.00
2DPCA	20	70.17
LDA	99	51.17
PCA	200	72.00

Table 7
Training time of 2DNPP and NPP for all tests (Sec).

	ORL	UMIST	AR (facial expressions)	AR (lighting conditions)	AR (occlusions)
2DNPP	0.96	0.18	0.08	0.25	0.52
NPP	407.04	251.57	172.36	253.31	212.97

2DLPP, 2DLDA, and 2DPCA. For ORL dataset as shown in Fig. 4, 2DLDA and 2DPCA work surprisingly well across all the values of *d*, which is the number of selected eigenvectors. 2DNPP performs slightly better than 2DLPP along with most of the values of *d* in this dataset. In UMIST, again 2DLDA performs better than other 2D methods observed from Fig. 5. 2DNPP outperforms 2DLPP across all the values of *d*. It is interesting to observe that 2DPCA achieves higher recognition rates when the value of *d* is larger than 7. The error rate curve obtained by 2DPCA becomes very smooth when the value of *d* is larger than 8. Figs. 6–8 illustrate the results in AR dataset under varying facial expressions, lighting conditions, and occlusions. Compared with Fig. 5, similar results are observed in Fig. 6 with respect to AR dataset under varying facial expressions. In Fig. 7 illustrating the results under different lighting conditions, 2DNPP outperforms other 2D methods across all the values of *d*. In contrast to Fig. 7, 2DNPP only achieves the best results along with the value of *d* varying from 16 to 19 as shown in Fig. 8.

6.4. Discussion

In general, from the preceding experiments, the 2DNPP algorithm designed in this paper delivers consistently promising

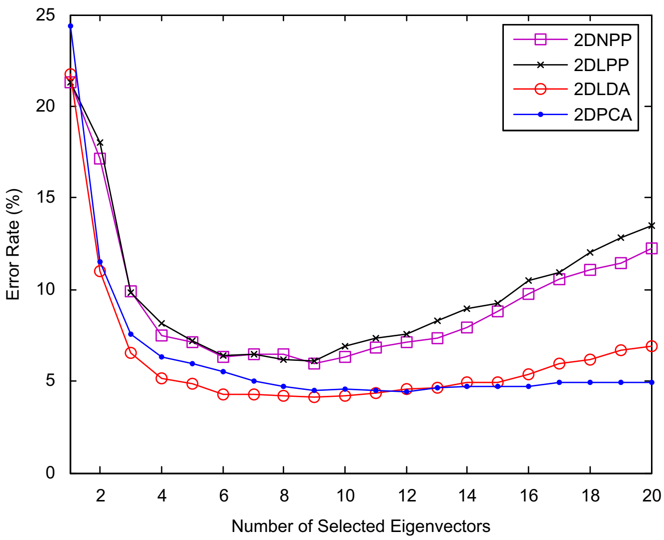

Fig. 4. Number of selected Eigenvectors against error rate for ORL dataset.

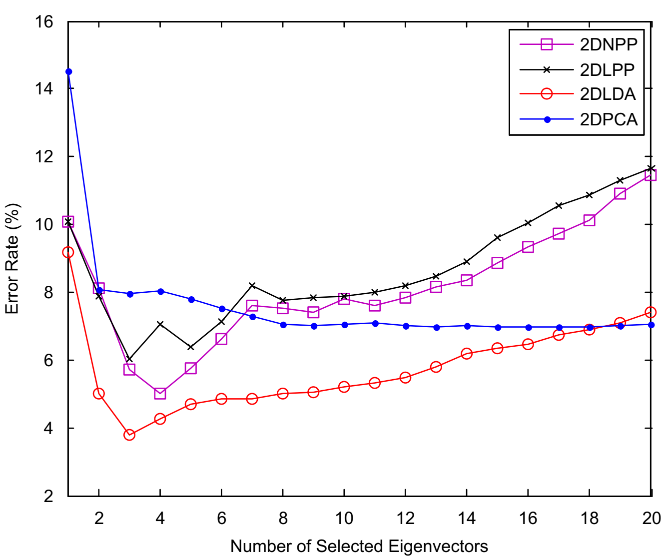

Fig. 5. Number of selected Eigenvectors against error rate for UMIST dataset.

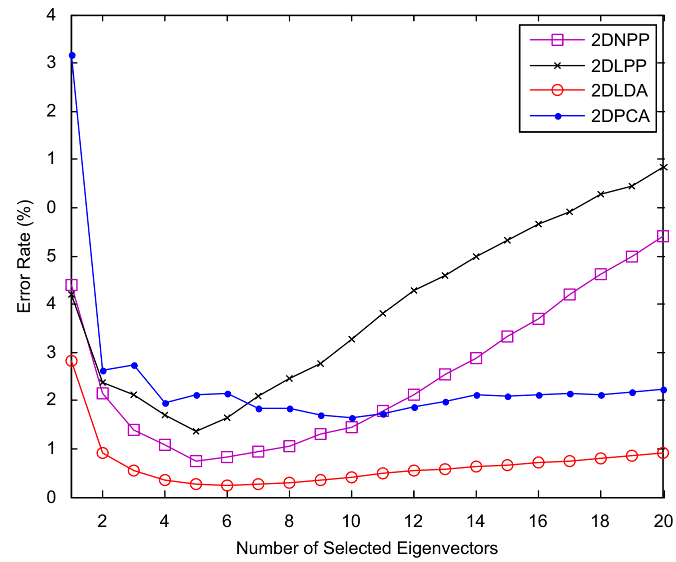

Fig. 6. Number of selected Eigenvectors against error rate for AR dataset under varying facial expressions.

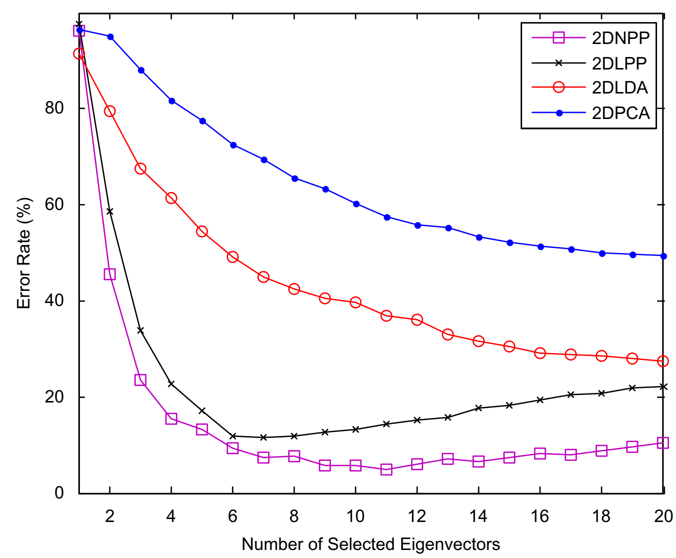

Fig. 7. Number of selected Eigenvectors against error rate for AR dataset under varying lighting conditions.

performance across all the experiments, especially for the faces under varying lighting conditions and occlusions. Specifically, we summarize the comparisons as follows:

(1) 2DNPP vs. 2DLPP: 2DNPP and 2DLPP employ different neighborhood weighting schemes, and they are equivalent under some special setting (see Section 4.1). But 2DNPP does not require any parameter selection during the neighborhood weighting, whilst 2DLPP is sensitive to the selection of the width of a Gaussian envelope [4]. From our experiments, we observe that 2DNPP consistently outperform 2DLPP. But we believe that 2DLPP under an appropriate parameter setting is comparable to 2DNPP. The parameter selection needs to be further investigated in the future work.

(2) 2DNPP vs. 2DLDA: From our observations, the 2DLDA approach is the most competing one to 2DNPP. 2DLDA works especially well for the face datasets with the variations in facial expressions and poses, for instance, the ORL and UMIST datasets. 2DLDA delivers only within 2% higher recognition rate than 2DNPP in ORL, UMIST, and AR (under varying facial expressions) datasets, but it works much worse than 2DNPP in AR dataset under varying lighting conditions. It indicates that for the local variations in faces such as occlusions and lighting conditions may degrade the performance of 2DLDA, because it only incorporates the discriminative information and the global geometry of data samples and lacks the local geometry description. In addition, 2DLDA is only implemented in a supervised setting, whilst the implementation of 2DNPP can be either unsupervised or supervised.

(3) 2DNPP vs. 2DPCA: 2DPCA is the origin of using 2D image inputs, and it has shown promising results compared to PCA in terms of recognition rate and time performance. But 2DPCA overlooks the locality description of samples. Experimental results suggest that 2DPCA performs slightly better than 2DNPP on the ORL dataset, which mainly shows variations in facial expressions, facial details, and poses. But 2DNPP significantly outperforms 2DPCA in particular on the AR dataset, which includes frontal views of faces in facial expressions, lighting conditions, and occlusions.

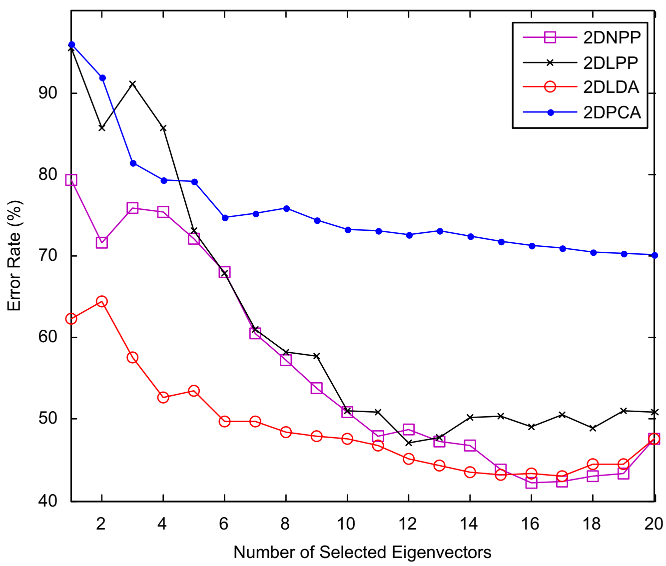

Fig. 8. Number of selected Eigenvectors against error rate for AR dataset under varying occlusions.

(4) 2DNPP vs. ONPP, NPP: 2DNPP is a straightforward extension of NPP for 2D feature inputs. ONPP is a variation of NPP without considering a condition of orthogonality on the projected data. From experimental results, we observe that 2DNPP consistently delivers better performance on recognition rate than ONPP and NPP. Meanwhile, the training of 2DNPP is over 400 times faster than ONPP and NPP.
(5) 2DNPP vs. OLPP, LPP: Both OLPP and LPP suffer the parameter selection issue with respect to the width of a Gaussian envelope as 2DLPP does. Despite OLPP delivers slightly better results than 2DNPP on the ORL dataset, 2DNPP outperforms OLPP and LPP in a significant rate in particular on the AR set. In addition, 2DNPP requires less training time. It is interesting to note that OLPP demonstrates promising results, in particular for the faces under the variations in facial expressions and poses, but it does not work well for the faces under varying lighting conditions and occlusions.
(6) 2DNPP vs. LDA, PCA: Both PCA and LDA work on 1D feature inputs. PCA aims at maximizing the covariance of training samples in an unsupervised manner, whilst LDA maximizes the class separability of a training set in a supervised manner. On the ORL dataset, the results of 2DNPP, LDA, and PCA are comparable. But 2DNPP outperforms LDA and PCA significantly on the AR dataset.
(7) 2DNPP vs. 2DONPP, 2DLPP vs. 2DOLPP: As we mentioned before, the difference between 2DNPP and 2DONPP, or 2DLPP and 2DOLPP is whether we impose a constraint to the objective function or not. From experimental results, it is surprisingly observed that 2DONPP and 2DOLPP almost have no capability of recognizing faces. We should note that ONPP and OLPP achieve promising results in comparison to NPP and LPP, respectively. This finding indicates a big necessity of the constraint to the objective function $\Phi(Y)$ in the form of $YY^T = I_d$ when it comes to dealing with two-dimensional (2D) feature inputs.

7. Conclusion

In this research, we design a 2DNPP algorithm, which is a two-dimensional extension of NPP algorithm. 2DNPP enables us to directly use a feature input of 2D image matrices rather than 1D vectors. We use the neighborhood weighting procedure, which is identical with that involved in NPP, to form the nearest neighbor affinity graph. We then present a detailed theoretical analysis of 2DNPP to build a relationship with 2DLPP [19], 2DPCA [17], and 2DLDA [18]. In line with the essence of the extension from 1D to 2D feature inputs, we extend the typical ONPP [4] to 2DONPP and make an extension of OLPP [4] to 2DOLPP as well. Extensive experimental verifications to evaluate the performance of 2DNPP on three well-known face image datasets show that 2DNPP consistently delivers promising performance, especially for the face datasets under the variations in lighting conditions and occlusions. Specifically, 2DNPP outperforms the standard NPP in a significant rate across all experiments with respect to recognition rate and training time. Likewise, 2DNPP achieves comparable performance compared with other competing methods. In this research, we find that the imposition of a constraint of the projection axes to the objective function in both NPP and LPP plays a crucial role for their 2D extensions in recognition. We need to investigate the theoretical support of this observation in the future work. According to [4], imposing the condition $YY^T = I_d$ leads to a criterion that is similar to that of PCA: the projected points y_i will tend to be different from one another because of the orthogonality of data projections. In other words, 2DNPP, 2DLPP, NPP, and LPP try to use the differentiation among the face images for recognition. In the future work, it will be interesting to investigate the performance of these algorithms in a data set with face images of very small difference. Motivated by [28,29], in the future work we may develop an efficient scheme to incorporate discriminative information to neighborhood preserving related methods such as 2DNPP, 2DLPP, ONPP, NPP, OLPP, and LPP.

Acknowledgment

The authors wish to thank reviewers for their detailed and useful comments. This work was supported in part by the Canada Chair Research Program and the Natural Sciences and Engineering Research Council of Canada.

References

[1] M. Turk, A.P. Pentland. Face recognition using eigenfaces, in: IEEE Conference Computer Vision and Pattern Recognition, 1991.
[2] P. Belhumeur, J. Hespanha, D. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, IEEE Transactions on Pattern Analysis and Machine Intelligence, Special Issue on Face Recognition 19 (7) (1997) 711–720.
[3] X. He, et al., Face recognition using laplacianfaces, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (3) (2005) 1–13.
[4] E. Kokiopoulou, Yousef Saad, Orthogonal neighborhood preserving projections: a projection-based dimensionality reduction technique, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (12) (2007) 2143–2156.
[5] I.T. Jolliffe, Principal Component Analysis, Springer-Verlag, New York, 1989.
[6] K. Fukunaga, Introduction to Statistical Pattern Classification, Academic Press, San Diego, California, USA, 1990.
[7] Zheng Wei-Shi, J.H. Lai, Z.Li. Stan, 1D-LDA vs. 2D-LDA: when is vector-based linear discriminant analysis better than matrix-based? Pattern Recognition 41 (2008) 2156–2172.
[8] Li-Fen Chen, et al., A new LDA-based face recognition system which can solve the small sample size problem, Pattern Recognition 33 (2000) 1713–1726.
[9] Jian Yang, Jing-Yu Yang, Why can LDA be performed in PCA transformed space? Pattern Recognition 36 (2003) 563–566.
[10] Xiaoxun Zhang, Yunde Jia, A linear discriminant analysis framework based on random subspace for face recognition, Pattern Recognition 40 (2007) 2585–2591.
[11] Daniel L. Swets, Juyang Weng, Using discriminant eigenfeatures for image retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (8) (1996) 831–836.
[12] J.B. Tenenbaum, V. de Silva, J.C. Langford., A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000) 2319–2323.
[13] M. Belkin, P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering, in: Proceedings Conference of Advances in Neural Information Processing Systems, 2001.
[14] Sam T. Roweis, K. Lawrence, Saul. nonlinear dimensionality reduction by locally linear embedding, Science 290 (2000) 2323–2326.
[15] T. Cox, M. Cox, Multidimensional Scaling, Chapman & Hall, London, 1994.

[16] X. He, P. Niyogi. Locality Preserving Projections, in: Proceedings Conference of Advances in Neural Information Processing Systems, 2003.
[17] J. Yang, D. Zhang, A.F. Frangi, and Jing-yu Yang. Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (1) (2004) 131–137.
[18] M. Li, B. Yuan., 2D-LDA: a statistical linear discriminant analysis for image matrix, Pattern Recognition Letters 26 (2005) 527–532.
[19] S. Chen, et al., 2D-LPP: a two-dimensional extension of locality preserving projections, Neurocomputing 70 (2007) 912–921.
[20] L. Saul, S. Roweis., Think globally, fit locally: unsupervised learning of nonlinear manifolds, Journal of Machine Learning Research 4 (2003) 119–155.
[21] F.R.K. Chung. Spectral graph theory, In: Proceedings of the Regional Conference Series in Mathematics, 1997, no. 92.
[22] P.J. Phillips. Support vector machines applied to face recognition, in: Proceedings of the Conference Advances in Neural Information Processing Systems, vol. 11, 1998, pp. 803–809.
[23] B. Moghaddam, A. Pentland, Probabilistic visual learning for object representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (1997) 696–710.
[24] H. Bischof, W. Schneider, A.J. Pinz, Multispectral classification of Landsat-images using neural networks, IEEE Transactions on Geoscience and Remote Sensing 30 (3) (1992) 482–490.
[25] F. Samaria, A. Harter. Parameterisation of a stochastic model for human face identification, in: Proceedings of the Second IEEE Workshop Applications of Computer Vision, 1994.
[26] D.B. Graham, N.M. Allinson, Characterizing Virtual eigensignatures for general purpose face recognition, Face Recognition: From Theory to Applications 163 (1998) 446–456.
[27] A.M. Martinez, R. Benavente. The AR Face Database. Technical Report CVC 24, 1998.
[28] E. Kokiopoulou, Y. Saad, Enhanced graph-based dimensionality reduction with repulsion Laplaceans, Pattern Recognition 42 (2009) 2392–2402.
[29] S. Yan, et al., Graph embedding and extensions: a general framework for dimensionality reduction, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (1) (2007) 40–51.
[30] J. Yang, et al., Two-dimensional discriminant transform for face recognition, Pattern Recognition 38 (7) (2005) 1125–1129.
[31] J. Ye, R. Janardan, Q. Li., Two-dimensional linear discriminant analysis, Advances in Neural Information Processing Systems 17 (2005) 1569–1576.

Haijun Zhang received his B.Eng. and M.Sc. from Northeastern University, Shenyang, P.R. China in 2004 and 2007, respectively. He worked as a research assistant at City University of Hong Kong in April–September 2007. He received his Ph.D. degree in the Department of Electronic Engineering, City University of Hong Kong, Hong Kong, in 2010. He is currently working as a Postdoctoral Fellow in the Department of Electrical and Computer Engineering, University of Windsor, ON, Canada. His research interests are data mining, machine learning, image processing, and their applications.

Q.M. Jonathan Wu (IEEE M'92–SM'09) received his Ph.D. degree in Electrical Engineering from the University of Wales, Swansea, UK, in 1990. From 1995, he worked at the National Research Council of Canada (NRC) for 10 years where he became a senior research officer and group leader. He is currently a Professor in the Department of Electrical and Computer Engineering at the University of Windsor, Canada. Dr. Wu holds the Tier1 Canada Research Chair (CRC) in Automotive Sensors and Sensing Systems. He has published more than 200 peer-reviewed papers in the areas of computer vision, image processing, intelligent systems, robotics, micro-sensors and actuators, and integrated micro-systems. His current research interests include 3D computer vision, active video object tracking and extraction, interactive multimedia, sensor analysis and fusion, and visual sensor networks. Dr. Wu is an Associate Editor for IEEE Transaction on Systems, Man, and Cybernetics (Part A). Dr. Wu has served on the Technical Program Committees and International Advisory Committees for many prestigious conferences.

Tommy W.S. Chow (IEEE M'93–SM'03) received the B. Sc. (First Hons.) and Ph.D. degrees from the University of Sunderland, Sunderland, U.K. He joined the City University of Hong Kong, Hong Kong, as a Lecturer in 1988. He is currently a Professor in the Electronic Engineering Department. His research interests include Neural Networks, Computational Intelligent, Machine learning, Pattern Recognition and applications.

Mingbo Zhao is currently pursuing the Ph.D. degree in the Department of Electronic Engineering, City University of Hong Kong, Hong Kong. He received his B.Eng. degree and Master Degree from the Department of Electronic Engineering, Shanxi University, Shanxi, P.R. China in 2005 and 2008, respectively. His current interests include data mining, machine learning, pattern recognition, and their applications.