Available online at www.sciencedirect.com
SCIENCE DIRECT
Neural Networks 18 (2005) 585–594
Neural Networks
www.elsevier.com/locate/neunet

2005 Special Issue

Generalized 2D principal component analysis for face image representation and recognition☆

Hui Kong[a,*], Lei Wang[a], Eam Khwang Teoh[a], Xuchun Li[a], Jian-Gang Wang[b], Ronda Venkateswarlu[b]

[a]School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang 639798, Singapore
[b]Division of Media, Institute for Infocomm Research, 119613 Singapore

Abstract

In the tasks of image representation, recognition and retrieval, a 2D image is usually transformed into a 1D long vector and modelled as a point in a high-dimensional vector space. This vector-space model brings up much convenience and many advantages. However, it also leads to some problems such as the Curse of Dimensionality dilemma and Small Sample Size problem, and thus produces us a series of challenges, for example, how to deal with the problem of numerical instability in image recognition, how to improve the accuracy and meantime to lower down the computational complexity and storage requirement in image retrieval, and how to enhance the image quality and meanwhile to reduce the transmission time in image transmission, etc. In this paper, these problems are solved, to some extent, by the proposed Generalized 2D Principal Component Analysis (G2DPCA). G2DPCA overcomes the limitations of the recently proposed 2DPCA (Yang et al., 2004) from the following aspects: (1) the essence of 2DPCA is clarified and the theoretical proof why 2DPCA is better than Principal Component Analysis (PCA) is given; (2) 2DPCA often needs much more coefficients than PCA in representing an image. In this work, a Bilateral-projection-based 2DPCA (B2DPCA) is proposed to remedy this drawback; (3) a Kernel-based 2DPCA (K2DPCA) scheme is developed and the relationship between K2DPCA and KPCA (Scholkopf et al., 1998) is explored. Experimental results in face image representation and recognition show the excellent performance of G2DPCA.

1. Introduction

In the tasks of image representation, recognition and retrieval, vector-space model may be the most popular one. It is adopted in most of the existing algorithms designed for these tasks. Under this model, the original two-dimensional (2D in short) image data are reshaped into a one-dimensional (1D in short) long vector, and then represented as a point in a high-dimensional vector space. This makes a great number of vector-space model based pattern recognition and analysis techniques be conveniently applied to image domain, and numerous successes have been achieved. However, it also leads to the following problems. Firstly, the intrinsic 2D structure of an image matrix is removed. Consequently, the spatial information stored therein is discarded and not effectively utilized. Secondly, each image sample is modelled as a point in such a high-dimensional space that a large number of training samples are often needed to get reliable and robust estimation about the characteristics of data distribution. It is known as the Curse of Dimensionality dilemma, which is frequently confronted in real applications. Thirdly, usually very limited number of data are available in real applications such as face recognition, image retrieval, and image classification. Consequently, Small Sample Size (SSS) problem (Fukunnaga, 1991) comes forth frequently in practice. The small sample size problem is defined as follows. When only t samples are available in an n-dimensional vector space with $t < n$, the sample covariance

☆ An abbreviated version of some portions of this article appeared in (Kong et al., 2005), as part of the IJCNN 2005 conference proceedings, published under the IEEE copyright.

* Corresponding author.
E-mail addresses: kongghui@pmail.ntu.edu.sg (H. Kong), pg03802060@ntu.edu.sg (H. Kong), elwang@ntu.edu.sg (L. Wang), eekteoh@ntu.edu.sg (E.K. Teoh), pg03454644@ntu.edu.sg (X. Li), jgwang@i2r.a-star.edu.sg (J.-G. Wang), vronda@i2r.a-star.edu.sg (R. Venkateswarlu).

0893-6080/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2005.06.041

matrix $\hat{\mathbf{C}}$ is calculated from the samples as

$$\hat{\mathbf{C}} = \frac{1}{t}\sum_{i=1}^{t}(\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T \quad (1)$$

where $\mathbf{m}$ is the mean of all the samples. $(\mathbf{x}_i - \mathbf{m})'$s are not linearly independent, because they are related by $\sum_{i=1}^{t}(\mathbf{x}_i - \mathbf{m}) = 0$. That is, $\hat{\mathbf{C}}$ is a function of $(t-1)$ or less linearly independent vectors. Therefore, the rank of $\hat{\mathbf{C}}$ is $(t-1)$ or less. This problem is often encountered in face recognition, image retrieval, and data mining tasks where t is very small but n is very large. Therefore, dimension reduction becomes one of the most important topics in these areas in pursuit of the low-dimensional representations of the original data with the requirement of minimum reconstruction error.

PCA is a well-established linear dimension-reduction technique. It finds the projection directions along which the reconstruction error to the original data is minimum, and projects the original data into a lower-dimensional space spanned by those directions corresponding to the top eigenvalues. Often, PCA is also known as the Karhunen-Löwe transformation. PCA has been widely used in many areas, such as face recognition, signal processing, and data mining etc. In image representation, Sirovich and Kirby originally used PCA to represent the human face images (Kirby & Sirovich, 1990; Sirovich & Kirby, 1987). In face recognition, Turk and Pentland proposed the well-known Eigenface (Turk & Pentland, 1991). Since then, PCA-based face/object recognition schemes have been investigated broadly. To deal with pose variation problem, Pentland et al. proposed the view-based and modular eigenspaces (Pentland et al., 1994). Murase and Nayar introduced the appearance manifolds (Murase & Nayar, 1995). To overcome the illumination variation problem, Bischof et al. (2004); Epstein et al. (1995); Hallinan (1994); Ramamoorthi (2002); Shashua (1992), and Zhao and Yang (1999), analyzed the ways of modelling the arbitrary illumination condition for PCA-based recognition methods. Recently, there is an increasing trend to investigate the kernel based PCA (KPCA) (Scholkopf et al., 1998) method. Another dimension-reduction method for face recognition is the fisher linear discriminant analysis (FLD) (Fukunnaga, 1991). FLD projects the data onto a lower-dimensional vector space such that the ratio of the between-class scatter to the within-class scatter is maximized, thus achieving maximum discrimination. The optimal projection (transformation) can be readily computed by solving a generalized eigenvalue problem. However, because of the SSS problem, the within-class covariance matrix, *Sw*, is singular so that the numerical problem is introduced in solving the optimal discriminating directions. To solve the singularity problem, the two-stage LDA was proposed (Belhumeur et al., 1997; Cevikalp et al., 2005; Swets & Weng, 1996; Zhao, 2000). Likewise, FLD is also extended to the kernel space in (Liu et al., 2002, 2003; Yang, 2002). Note that all the above techniques adopt the vector-space model and transform a 2D image matrix into a long vector by concatenating the column or row vectors therein. Hence, they are inevitably affected by the problems of curse of Dimensionality and Small Sample Size.

Recently, Two-Dimensional Principal Component Analysis (2DPCA) (Yang et al., 2004), a variant of the classical PCA, is developed for face recognition as another linear image projection technique. Different from the classical PCA, 2DPCA takes a 2D matrix based representation model rather than simply the 1D vector based one. When performing 2DPCA, the original 2D image matrix does not need to be converted as a long vector beforehand. Instead, a covariance matrix is constructed by using the 2D image matrices directly. The projection directions are computed based on this covariance matrix to guide principal component analysis. As reported in (Yang et al., 2004), 2DPCA can achieve better performance than PCA in face recognition when the number of samples is small. However, there still remains several problems in 2DPCA. Firstly, the authors did not explicitly explain the reason why 2DPCA can achieve a better performance than PCA. Secondly, the existing reported 2DPCA adopts a unilateral-projection (right-multiplication) scheme only, and the disadvantage arising in this way is that more coefficients are needed to represent an image in 2DPCA than in PCA. This means a lower compression rate in representing an image. Thirdly, 2DPCA is still a linear projection technique, which cannot effectively deal with the higher-order statistics among the row/column vectors of an image. However, it is well known that the object/face appearances often lie in a nonlinear low-dimensional manifold when there exist pose or/and illumination variations (Murase & Nayar, 1995). The linear 2DPCA is not able to model such a nonlinearity, and this prevents it from higher recognition rate.

To remedy these drawbacks in the existing 2DPCA, this paper proposes a framework of Generalized 2D Principal Component Analysis (G2DPCA), which is more useful and efficient for real applications. G2DPCA extends the standard 2DPCA from the following three perspectives: firstly, the essence of 2DPCA is studied in theoretical sense and the relationship between 2DPCA and PCA are exposed. These give rise to an explicit explanation of the reason why 2DPCA can often achieve better performance than PCA.

Secondly, instead of a unilateral-projection scheme, a bilateral-projection based 2DPCA (B2DPCA) is developed. There, two sets of projection directions are constructed simultaneously, and are used to project the row and column vectors of the image matrices to two different subspaces, respectively. The advantage of B2DPCA over 2DPCA is that an image can be

effectively represented with much less number of coefficients, achieving a higher compression rate. Thirdly, to model the nonlinear structures which are often presented in practical face recognition tasks, the kernel trick is incorporated in the linear method and a Kernel-based 2DPCA (K2DPCA) is derived. It can effectively remedy the drawback of 2DPCA in modeling the nonlinear manifold in face images. A preliminary work of this paper is presented in (Kong et al., 2005).

The remainder of this paper is organized as follows: 2DPCA algorithm is reviewed in Section 2. The essence of 2DPCA and the relationship between 2DPCA and PCA are revealed in Section 3. B2DPCA algorithm and the image reconstruction method using B2DPCA are developed in Section 4. The Kernel based 2DPCA is introduced in Section 5. Experimental results are presented in Section 6. We draw the conclusions in the last section.

2. 2D principal component analysis

Let $\mathbf{x}$ be an n-dimensional unitary column vector. The idea is to project image $\mathbf{A}$, an $m\times n$ matrix, onto $\mathbf{x}$ by $\mathbf{y}=\mathbf{A}\mathbf{x}$. To determine the optimal projection vector $\mathbf{x}$, the total scatter of the projected samples, $\mathbf{S}_x$, is used to measure the goodness of $\mathbf{x}$. $\mathbf{S}_x=\mathbf{x}^T\ E\{[\mathbf{A}-E(\mathbf{A})]^T[\mathbf{A}-E(\mathbf{A})]\}\mathbf{x}=\mathbf{x}^T\mathbf{S}_A\mathbf{x}$, where $\mathbf{S}_A=E\{[\mathbf{A}-E(\mathbf{A})]^T[\mathbf{A}-E(\mathbf{A})]\}$, called the image covariance matrix. Suppose that there are totally M training samples $\{\mathbf{A}_i\}$, $i=1,2,\ldots,M$, and the average image is denoted by $\bar{\mathbf{A}}$, then $\mathbf{S}_A=\frac{1}{M}\sum_{i=1}^{M}[\mathbf{A}_i-\bar{\mathbf{A}}]^T[\mathbf{A}_i=\bar{\mathbf{A}}]$. The optimal projection direction, $\mathbf{x}_{\mathrm{Opt}}$, is the eigenvector of $\mathbf{S}_A$ corresponding to the largest eigenvalue. Usually a set of orthonormal projection directions, $\mathbf{x}_1, \mathbf{x}_2,\ldots, \mathbf{x}_d$, are selected and these projection directions are the orthonormal eigenvectors of $\mathbf{S}_A$ corresponding to the first d largest eigenvalues. For a given image $\mathbf{A}$, let $\mathbf{y}_k=\mathbf{A}\mathbf{x}_k$, $k=1,2,\ldots,d$. A set of projected feature vectors $\mathbf{y}_k$, the principal components (vectors) of $\mathbf{A}$, are obtained. Then the feature matrix of $\mathbf{A}$ is formed as $\mathbf{B}=[\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_d]$. The nearest-neighborhood classifier is adopted for classification. The distance between two arbitrary feature matrices, $\mathbf{B}_i$ and $\mathbf{B}_j$, is defined as $d(\mathbf{B}_i,\mathbf{B}_j)=\sum_{k=1}^{d}\|\mathbf{y}_k^i-\mathbf{y}_k^j\|_2$, where $\|\mathbf{y}_k^i-\mathbf{y}_k^j\|_2$ is the Euclidean distance between $\mathbf{y}_k^i$ and $\mathbf{y}_k^j$.

3. The essence of 2DPCA

The work by Yang et al. experimentally shows that 2DPCA can achieve better performance in face recognition. However, the essence of 2DPCA and its relationship to PCA is not discussed in (Yang et al., 2004). We believe that this discussion is indispensable for understanding the intrinsic mechanism of 2DPCA and its advantages over PCA. The following work in this paper will theoretically explain the essence of 2DPCA and also its relationship to PCA.

Theorem 1. 2DPCA, performed on the 2D images, is essentially PCA performed on the rows of the images if each row is viewed as a computational unit.

Proof: Let $\mathbf{A}_i$ be the i-th training sample, $\mathbf{A}_i^j$ be the j-th row of $\mathbf{A}_i$. Let $E(\mathbf{A})$ be the mean of all training samples, $E(\mathbf{A})^j$ be the j-th row of $E(\mathbf{A})$. Let $\hat{\mathbf{A}}_i^j$ be the centered $\mathbf{A}_i$ and $\hat{\mathbf{A}}_i^i$ be the centered $\mathbf{A}_i^j$, where $\hat{\mathbf{A}}_i=\mathbf{A}_i-E(\mathbf{A})$ and $\hat{\mathbf{A}}_i^j=\mathbf{A}_i^j-E(\mathbf{A})^j$.

Because of the limited number of available samples in specific applications, $\mathbf{S}_A$ is often estimated by:

$$\mathbf{S}_A=\frac{1}{M}\sum_{i=1}^{M}[\mathbf{A}_i-E(\mathbf{A})]^T[\mathbf{A}_i-E(\mathbf{A})] \tag{2}$$

It can also be written as,

$$\mathbf{S}_A=\frac{1}{M}\Psi\Psi^T \tag{3}$$

where

$$\Psi=\left[[\mathbf{A}_1-E(\mathbf{A})]^T,\cdots,[\mathbf{A}_M-E(\mathbf{A})]^T\right] \tag{4}$$

or

$$\Psi=\left[\left[(\hat{\mathbf{A}}_1^1)^T,\ldots,(\hat{\mathbf{A}}_1^m)^T\right],\ldots,\left[(\hat{\mathbf{A}}_M^1)^T,\ldots,(\hat{\mathbf{A}}_M^m)^T\right]\right] \tag{5}$$

Therefore, $\mathbf{S}_A$ can be viewed as the covariance matrix evaluated using the rows of all the centered training samples. In 2DPCA, the maximization of $\mathbf{S}_x$ is equal to maximize $\mathbf{x}2^T\Psi\Psi^T\mathbf{x}$. This translates into the eigen-analysis of $\Psi\Psi^T$:

$$\lambda_i\mathbf{x}_i=\Psi\Psi^T\mathbf{x}_i \tag{6}$$

Hence, 2DPCA performed on the image matrices is essentially the PCA performed on the rows of all the images. □

So far, we can give the explanation of the advantages of 2DPCA over PCA. Firstly, as the dimension of the row vectors in an image is much smaller than that of the long vector transformed from the entire image, the dilemma of curse of dimensionality diminishes. Secondly, as the input feature vectors to be analyzed are actually the row vectors of the training images, the feature set is significantly enlarged. Therefore, the SSS problem does not exist in 2DPCA any more. Thirdly, the 2D spatial information is well preserved by using the original 2D image matrix rather than reshaping it to a long vector. Fourthly, the distance function adopted in the classification criterion of 2DPCA is a global combination of all the local Eigen-feature distances. In terms of the first two advantages, it can be known that the covariance matrix in 2DPCA can be estimated more robustly and accurately than that in PCA. Although this is also noticed by (Yang et al., 2004), it did not explore the intrinsic reasons mentioned above.

4. Bilateral 2d principal component analysis

As mentioned in Section 1, 2DPCA is a unilateral-projection-based scheme, where only right multiplication is taken. Referring to the above analysis that 2DPCA is essentially PCA performed on the row vectors of all the available images, we know that a unilateral scheme will have the correlation information among the column vectors of the images lost. Compared with PCA, a disadvantage of the unilateral-projection scheme is that more coefficients are needed to represent an image. To remove these problems, a bilateral-projection scheme is taken instead, and a bilateral-projection-based 2DPCA (B2DPCA) is proposed in this section. Compared with the existing 2DPCA, B2DPCA can effectively remove the redundancies among both rows and columns of the images and thus lower down the number of coefficients used to represent an image. Also, the correlation information in both rows and columns of the images are considered in B2DPCA, and this will benefit the subsequent classification performed in the obtained subspaces.

4.1. Algorithm

Let $\mathbf{U} \in \mathcal{R}^m \times \mathcal{R}^l$ and $\mathbf{V} \in \mathcal{R}^n \times \mathcal{R}^r$ be the left- and right-multiplying projection matrix, respectively. It is assumed that all the samples are all centered in the later sections. For an $m \times n$ image $\mathbf{A}_i$ and an $l \times r$ projected image $\mathbf{B}_i$, the bilateral projection is formulated as follows:

$$\mathbf{B}_i = \mathbf{U}^T \mathbf{A}_i \mathbf{V} \tag{7}$$

where $\mathbf{B}_i$ is the extracted feature matrix for image $\mathbf{A}_i$.

The common optimal projection matrices, $\mathbf{U}_{\text{Opt}}$ and $\mathbf{V}_{\text{Opt}}$ in Eq. (7) can be computed by solving the following minimization problem such that $\mathbf{U}_{\text{Opt}} \mathbf{B}_i \mathbf{V}_{\text{Opt}}^T$ gives the best approximation of $\mathbf{A}_i$, $i = 1, \ldots, M$:

$$\left[\mathbf{U}_{\text{Opt}}, \mathbf{V}_{\text{Opt}}\right] = \arg\min \sum_{i=1}^{M} \|\mathbf{A}_i - \mathbf{U}\mathbf{B}_i\mathbf{V}^T\|_F^2 \tag{8}$$

where M is the number of data samples and $\|\cdot\|_F$ is the Frobenius norm of a matrix.

Theorem 2. The minimization of Eq. (8) is equivalent to the maximization of $\sum_{i=1}^{M} ||\mathbf{U}^T \mathbf{A}_i \mathbf{V}||_F^2$.

The proof is given in Appendix A.

Given the data set $\mathbf{A}_i \in \mathrm{R}^m \times \mathrm{R}^n$, $i = 1, \ldots, M$, the covariance matrix of the projected samples is defined as:

$$\mathbf{C} = \frac{1}{M} \sum_{i=1}^{M} \mathbf{B}_i^T \mathbf{B}_i \tag{9}$$

where $\mathbf{B}_i$ is defined in Eq. (7). By replacing $\mathbf{B}_i$ with $\mathbf{U}^T \mathrm{A}_i \mathbf{V}$, it translates into:

$$\mathbf{C} = \frac{1}{M} \sum_{i=1}^{M} (\mathbf{U}^T \mathbf{A}_i \mathbf{V})^T (\mathbf{U}^T \mathbf{A}_i \mathbf{V}) \tag{10}$$

and it is trivial to check that $tr(\mathbf{C}) = \frac{1}{M} \sum_{i=1}^{M} ||\mathbf{U}^T \mathbf{A}_i \mathbf{V}||_F^2$.

In this regard, maximizing the trace of the covariance matrix of the projected samples is equivalent to maximizing $\sum_{i=1}^{M} ||\mathbf{U}^T \mathbf{A}_i \mathbf{V}||_F^2$, while maximizing $\sum_{i=1}^{M} ||\mathbf{U}^T \mathbf{A}_i \mathbf{V}||_F^2$ has been shown to be equivalent to minimizing $\sum_{i=1}^{M} ||\mathbf{A}_i - \mathbf{U}\mathbf{B}_i\mathbf{V}^T||_F^2$ and optimally reconstructing (approximating) the images. Therefore, the proposed bilateral-projection scheme is consistent with the principle of PCA and 2DPCA, and it can be viewed as a generalized 2DPCA, i.e. the standard 2DPCA is a special form of the bilateral 2DPCA.

To our knowledge, there is no close-form solution for the maximization of $\sum_{i=1}^{M} ||\mathbf{U}^T \mathbf{A}_i \mathbf{V}||_F^2$ because $\mathbf{C} = \frac{1}{M} \sum_{i=1}^{M} \mathbf{V}^T \mathbf{A}_i^T \mathbf{U}$ $\mathbf{U}^T \mathbf{A}_i \mathbf{V}$ and there is no direct eigen decomposition for such a coupled covariance matrix. Considering this, an iterative algorithm is proposed to compute $\mathbf{U}_{\text{Opt}}$ and $\mathbf{V}_{\text{Opt}}$. Before we give details of the iterative algorithm, we have the following two Lemmas.

Lemma 1. Given the $\mathbf{U}_{\text{Opt}}$, $\mathbf{V}_{\text{Opt}}$ can be obtained as the matrix formed by the first r eigenvectors corresponding to the first r largest eigenvalues of $\mathbf{C}_{\text{v}} = \frac{1}{M} \sum_{i=1}^{M} \mathbf{A}_i^T \mathbf{U}_{\text{Opt}} \mathbf{U}_{\text{Opt}}^T \mathbf{A}_i$.

Proof: Since $\mathbf{U}_{\text{Opt}}$ and $\mathbf{V}_{\text{Opt}}$ maximize $tr(\mathbf{C})$, which equals $tr\left(\frac{1}{M} \sum_{i=1}^{M} \mathbf{V}^T \mathbf{A}_i^T \mathbf{U}\mathbf{U}^T \mathbf{A}_i \mathbf{V}\right)$. If $\mathbf{U}_{\text{Opt}}$ is known, $tr(\mathbf{C}) = tr\left(\frac{1}{M} \sum_{i=1}^{M} \mathbf{V}^T \mathbf{A}_i^T \mathbf{U}_{\text{Opt}} \mathbf{U}_{\text{Opt}}^T \mathbf{A}_i \mathbf{V}\right) = tr(\mathbf{V}^T \mathbf{C}_v \mathbf{V})$.

Therefore, the maximization of $tr(\mathbf{C})$ equals to solve the first r eigenvectors of $\frac{1}{M} \sum_{i=1}^{M} \mathbf{A}_i^T \mathbf{U}_{\text{Opt}} \mathbf{U}_{\text{Opt}}^T \mathbf{A}_i$ corresponding to the first r largest eigenvalues. □

Lemma 2. Given $\mathbf{V}_{\text{Opt}}$, $\mathbf{U}_{\text{Opt}}$ can be obtained as the matrix formed by the first l eigenvectors corresponding to the first l largest eigenvalues of $\mathbf{C}_u = \frac{1}{M} \sum_{i=1}^{M} \mathbf{A}_i V_{\text{Opt}} \mathbf{U}_{\text{Opt}}^T \mathbf{A}_i^T$.

The proof of Lemma 2 is similar to that of Lemma 1. □

Table 1
The algorithm for computing $\mathbf{U}_{\text{opt}}$ and $\mathbf{V}_{\text{opt}}$

S_1	Initialize $\mathbf{U}$, $\mathbf{U} = \mathbf{U}_0$ and $i = 0$
S_2	While not convergent
S_3	Compute $\mathbf{C}_v$ and the eigenvectors $\{e_j^V\}_{j=1}^r$ corresponding to its top eigenvalues, then $\mathbf{V}_i \leftarrow [e_1^V, \ldots, e_r^V]$
S_4	Compute C_u and the eigenvectors $\{e_j^U\}_{j=1}^l$ corresponding to its top eigenvalues, then $\mathbf{U}_i \leftarrow [e_1^U, \ldots, e_l^U]$
S_5	$i \leftarrow i + 1$
S_6	End while
S_7	$\mathbf{v}_{\text{Opt}} \leftarrow \mathbf{v}_{\text{i-1}}$ and $\mathbf{U}_{\text{Opt}} \leftarrow \mathbf{U}_{\text{i-1}}$
S_8	Feature extraction: $\mathbf{B}_i = \mathbf{U}_{\text{Opt}}^T \mathbf{A}_i \mathbf{V}_{\text{Opt}}$

Fig. 1. Ten sample images of two subjects in ORL database.

Fig. 2. Eighteen sample images of subject 1a from UMIST face database labelled by #1, #2,…, # 18 from left to right.

By Lemma 1 and 2, the detailed iterative algorithm to compute $\mathbf{U}_{\text{Opt}}$ and $\mathbf{V}_{\text{Opt}}$ is listed in Table 1. Theoretically, the solutions are local optimal because the solutions are dependent on the initialization of $\mathbf{U}_0$. By extensive experiments, $\mathbf{U}_0 = \mathrm{I}_m$, a setting we adopted, will produce excellent results. Another issue that deserves attention is the convergency. We consider the mean reconstruction error, i.e.

$$\mathcal{E} = \frac{1}{M}\sum_{i=1}^{M} \|\mathbf{A}_i - \mathbf{U}\mathbf{B}_i\mathbf{V}^T\|_F \tag{11}$$

We use the relative reduction of $\mathcal{E}$ value to check the convergence of B2DPCA. More specifically, let $\mathcal{E}(i)$ and $\mathcal{E}(i-1)$ be the error at the i-th and $(i-1)$-th iteration, respectively. The convergence of this algorithm can be judged by whether it can satisfy the following inequity.

$$\frac{\mathcal{E}(i-1)-\mathcal{E}(i)}{\mathcal{E}(i-1)} \le \mu \tag{12}$$

where μ is a small positive number. Our experiments in the later section will show that the iterative algorithm usually converges within two iterations.

4.2. Images representation and reconstruction using B2DPCA

Since we have obtained the common optimal projection matrices, $\mathbf{U}_{\text{Opt}}^T \in \mathcal{R}^m \times \mathcal{R}^l$ and $\mathbf{V}_{\text{Opt}} \in \mathcal{R}^n \times \mathcal{R}^r$, for any image $\mathbf{A}_i \in \mathcal{R}^m \times \mathcal{R}^n$, its feature matrix $B_i \in \mathcal{R}^l \times \mathcal{R}^r = \mathbf{U}_{\text{Opt}}^T\mathbf{A}_i\mathbf{V}_{\text{Opt}}$. Therefore, $\mathbf{B}_i$ is the coefficient matrix that can be used to reconstruct the image $\mathbf{A}_i$ by $\hat{\mathbf{A}}_i = \mathbf{U}_{\text{Opt}}\mathbf{B}_i\mathbf{V}_{\text{Opt}}^T$.

5. Kernel based 2d principal component analysis

Kernel Principal Component Analysis (KPCA) is a generalized version of PCA. In KPCA, through the kernel trick, the input data are mapped onto a higher- or even infinite-dimensional space and PCA is performed therein. The kernel trick achieves this mapping implicitly and incurs very limited computational overhead. More important, incorporating the kernel trick helps to capture the higher order statistical dependencies among the input data. KPCA has been applied to face recognition and it has demonstrated better performance than PCA. Likewise, the kernelization of 2DPCA will give a great help to model the nonlinear structures in the input data. Similar to KPCA, a nonlinear mapping without explicit function is performed. Different from KPCA, this mapping is performed on each row of all the image matrices, i.e. let $\Phi : \mathbf{R}^t \to \mathbf{R}^f$, $f > t$, be the mapping on each row of the image, where t is the length of the rows of an image and f can be arbitrarily large. The dot product in the feature space of $\mathbf{R}^f$ can be conveniently calculated via a predefined kernel function, such as the commonly used Gaussian RBF kernel.

For convenience, it is assumed that all the mapped data are centered by the method in (Scholkopf et al., 1998). Let $\hat{\Phi}(\mathbf{A}_i)$ be the i-th mapped image in which $\hat{\Phi}(\mathbf{A}_i^j)$ be the j-th centered row vector of it. The covariance matrix $\mathbf{C}^{\Phi}$ in $\mathbf{R}^f$:

$$\mathbf{C}^{\Phi} = \frac{1}{M}\sum_{i=1}^{M} \hat{\Phi}(\mathbf{A}_i)^T\hat{\Phi}(\mathbf{A}_i) \tag{13}$$

where

$$\hat{\Phi}(\mathbf{A}_i) = [\hat{\Phi}(\mathbf{A}_i^1)^T, \hat{\Phi}(\mathbf{A}_i^2)^T, \ldots, \hat{\Phi}(\mathbf{A}_i^m)^T]^T$$

and m is the number of row vectors. If $\mathbf{R}^f$ is infinite-dimensional, $\mathbf{C}^{\Phi}$ is inf×inf in size. It is intractable to

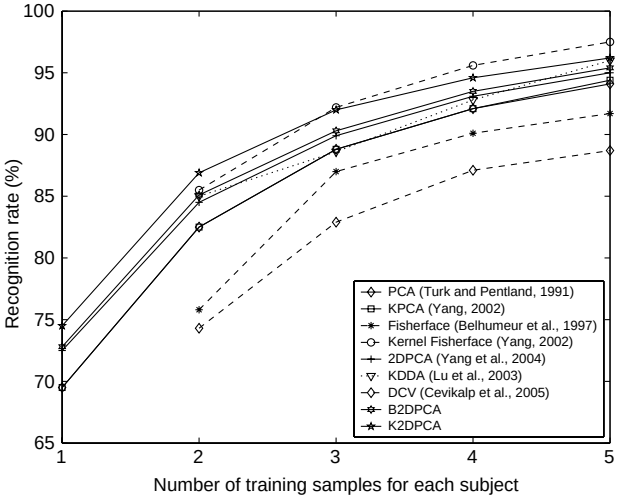

Fig. 3. Experimental comparison (%) on ORL database.

Fig. 4. Sample images of one subject from Yale face database.

directly calculate the eigenvalues, λ_i, and the eigenvectors, $\mathbf{v}_i$, that satisfy

$$\lambda_i \mathbf{v}_i = \mathbf{C}^{\Phi} \mathbf{v}_i \tag{14}$$

However, K2DPCA can be implemented using KPCA according to the following theorem.

Theorem 3. The above defined kernelized 2DPCA on the images is essentially KPCA performed on the rows of all the training image matrices if each row is viewed as an computational unit.

The proof is given in Appendix B.

After projecting each mapped row vector of all the training and test images onto the first d reserved eigenvectors in the feature space, an $m \times d$ feature matrix is obtained for each image. The nearest-neighborhood classifier is then adopted for classification whose steps are similar to 2DPCA.

6. Experimental results and discussions

6.1. Face recognition on ORL, UMIST and Yale databases

The proposed B2DPCA and K2DPCA methods are applied to the face image reconstruction and recognition. They are evaluated on three well-known face databases: ORL, UMIST and Yale databases. ORL contains images from 40 individuals, each providing 10 different images. The pose, expression and facial details (e.g. with glasses or without glasses) variations are also included. The images are taken with a tolerance for some tilting and rotation of the face of up to 20°. Moreover, there are also some variations in the scale of up to about 10%. Ten sample images of two persons from the ORL database are shown in Fig. 1. UMIST consists of 564 images of 20 people with large pose variations. In our experiment, 360 images with 18 samples for each subject are used to ensure that the face appearance changes from profile to frontal orientation with a step of 5° separation (labelled from 1 to 18). The sample images for subject 1 are shown in Fig. 2. Yale contains altogether 165 images for 15 subjects. There are 11 images per subject, one for each of the following facial expressions or configurations: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink.

All images in ORL, UMIST and Yale databases are grayscale and normalized to a resolution of 56×46 pixels. The ORL database is employed to check whether the proposed methods have good generalization ability under the circumstances that the pose, expression, and face scale variations exist concurrently. The UMIST face database is used to examine the performance when face orientation varies significantly. The Yale face database is used to see whether the proposed algorithms can achieve good result when there exist occlusion, expression and illumination variations.

To test the recognition performance with respect to different number of training samples on ORL, k ($1 \le k \le 5$) images of each subject are randomly selected for training and the remaining $(10-k)$ images for testing. When $2 \le k \le 5$, 50 times of random selections are performed. When k equals 1, there are 10 possible selections for training. The final recognition rate is the average of all. The performance of B2DPCA and K2DPCA compared with that of the current methods is listed in Fig. 3.

To test the recognition performance with respect to different number of training samples on Yale, only nine images of each person are used (the two images with left-light and right-light are excluded). The nine sample images for one of the subjects in Yale are shown in Fig. 4. k ($1 \le k \le 5$) images of each subject are randomly selected for training and the remaining $(9-k)$ for test. When $2 \le k \le 5$, 50 times of random selections are performed. When k equals 1, there are nine possible selections for training. The final recognition rate is the average of all. The performance is listed in Fig. 5. Two

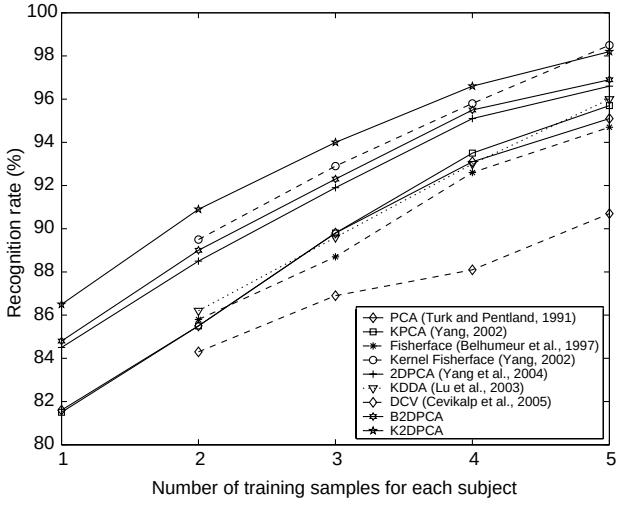

Fig. 5. Experimental comparison (%) on Yale database.

Table 2
Experiment results (%) OMIST database

	#5, #14	#1, #7, #13	#2, #8, #14	#3, #9, #15	#4, #10, #16	#5, #11, #17	#6, #12, #18
PCA (Turk & Pentland, 1991)	80.3	82.7	89.7	90.7	90.7	88.0	86.0
KPCA (Yang, 2002)	80.9	86.0	87.0	91.0	92.0	89.3	87.3
LDA (Belhumeur et al., 1997)	77.5	90.0	91.3	95.0	96.3	94.3	91.7
Kernel Fisherface (Yang, 2002)	9.5	94.7	96.7	98.3	99.0	98.0	97.3
2DPCA (Yang et al., 2004)	90.3	91.0	93.0	95.0	95.0	93.7	92.3
KDDA (Lu et al., 2003)	87.8	94.0	96.0	95.7	97.3	95.7	95.7
DCV (Cevikalp et al., 2005)	84.1	89.7	93.7	97.7	94.7	92.7	88.0
B2DPCA	90.7	91.7	93.4	95.3	95.8	94.0	92.8
K2DPCA	92.7	94.0	94.3	95.7	97.0	95.7	94.0

experiments, with small number of training samples (two and three), are conducted on UMIST database. When the number of training samples for each individual is two, we select the {#5, #14} face images of each subject for training, the remaining for test. When the number of training samples is three for each subject, six groups are selected for training, i.e. 1{#1,#7,#13}, 2{#2,#8,#14}, 3{#3,#9,#15}, 4{#4,#10,#16}, 5{#5,#11,#17} and 6{#6,#12,#18}. The remaining images corresponding to each group are used for test. The performance of B2DPCA and K2DPCA is compared with that of the state-of-the-art methods in Table 2.

The Gaussian RBF kernel is adopted in K2DPCA, the optimal results are obtained when the width, δ, of the kernel is about 2.72. The optimal dimensions of $\mathbf{U}_{\text{opt}}$ and $\mathbf{V}_{\text{opt}}$ of B2DPCA in both experiments are around 56×5 and 56×5, therefore, the size of the extracted feature matrix for each image is 5×5. For both experiments, the nearest-neighborhood classification criterion is adopted and the distance between any two feature matrices is the same as the one used in 2DPCA. Through experiments, we find that B2DPCA is better than 2DPCA, K2DPCA does outperform 2DPCA and KPCA as explained in Section 5.

It should be pointed out that FLD is good at discrimination rather than representation. FLD can generally achieve better performance than PCA under noticeable illumination and pose variations. However, FLD will be inferior to PCA if the illumination and pose variations are not significant and there are very limited training samples for each subject. The reason for this lies in two-fold: firstly, when there are large pose- and illumination-variations in face images, the top eigenvectors in PCA-based approaches does not model identity information but these external variations. Secondly, in FLD, the null space of $\mathbf{S}_w$, whose rank is $C-1$, is discarded. When the number of training samples for each subject is small (e.g. 2), the rank of null space of $\mathbf{S}_w$ is comparable to the rank of range space of $\mathbf{S}_w$. Therefore, discarding the whole null space of $\mathbf{S}_w$ will lead to a loss of a large quantity of discriminant information. However, with the number of training samples increasing, the rank of null space of $\mathbf{S}_w$ is much smaller than the rank of range space of $\mathbf{S}_w$ and discarding the whole null space of $\mathbf{S}_w$ will lose relatively little useful information.

We also find that K2DPCA is superior to Fisherface (FLD) and DCV. Additionally, K2DPCA is comparable to KDDA in all the experiments we have done, and it is better than KDDA when the number of training samples is 2, 3 and 4. K2DPCA is even better than Kernel Fisherface method when the number of training sample is 2. K2DPCA is better than B2DPCA in generalization ability.

6.2. The effect of d-value on recognition performance

A common d is set to be the same for both I and r in B2DPCA, therefore, the final feature image obtained from B2DPCA for each image is a $d\times d$ square matrix. A large d will result in a small compression rate while a small d will lose some important information for classification. To illustrate this situation, lots of experiments are conducted on two databases. The results are shown in Fig. 6, where the x-axis denotes the d-value and the y-axis denotes the recognition rate. Three experiments with different number of training samples (2, 3 and 4, respectively) for each subject are done on ORL database. Three experiments with different

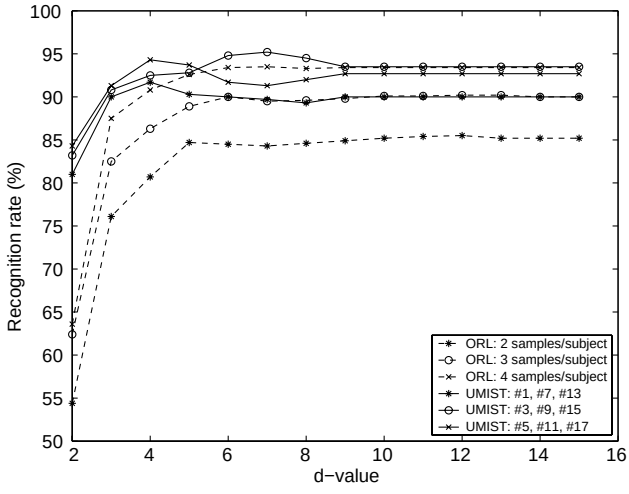

Fig. 6. The effect of different d-value on recognition rate of B2DPCA.

Fig. 7. First row: raw images. Second row and fourth row: image reconstructed and compressed by 2DPCA using 2 and 8 principal component (vectors), respectively. Third row and fifth row: image reconstructed and compressed by B2DPCA with $d=10$ and $d=20$, respectively.

training set (l{#1,#7,#13}, 3{#3,#9,#15}, 5{#5,#11,#17}) are conducted on UMIST. From Fig. 6, when the d-value is about 5, B2DPCA will achieve the highest recognition rate. When d is larger, the recognition rate is nearly constant. Meantime, to ensure an efficient classification and high compression rate, d is therefore set to be 5.

6.3. Face image reconstruction and compression

2DPCA is an excellent dimension-reduction tool for image processing, compression, storage and transmission. In this part, we compare the compression rate and reconstruction effect of B2DPCA with that of 2DPCA. Fig. 7 shows the reconstruction effect of them, where the raw images lie in the first row and the reconstructed image by 2DPCA using 2 and 8 principal component (vectors) are shown in the second and fourth rows, respectively.]The reconstructed images by B2DPCA with $d=10$ and $d=20$ are shown in the third and fifth rows. Therefore, the second and third rows have almost the same compression rate since $(56\times46/56\times2)\approx(56\times46/10\times10)$, while the fourth and fifth rows have almost the same compression rate since $(56\times46/56\times8)\approx(56\times46/20\times20)$. But the effect of the reconstruction by B2DPCA in the third and fifth rows are much better than that by 2DPCA in the second and fourth rows, respectively.

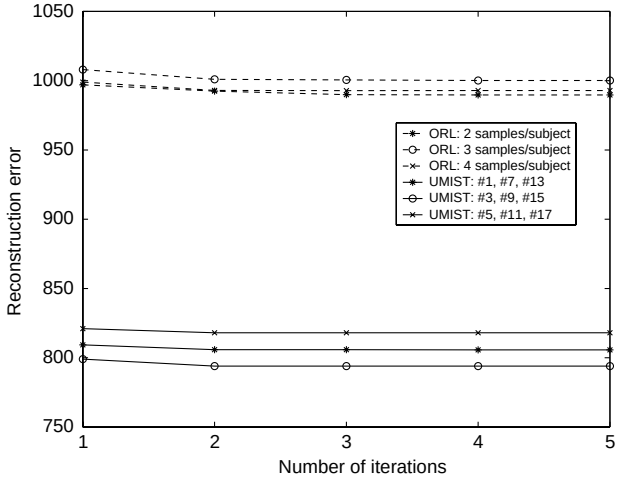

Fig. 8. Convergence of B2DPCA.

6.4. Convergence of B2DPCA

The image reconstruction error can be used as a measure of the convergency of B2DPCA algorithm. In this experiment, the reconstruction error is shown as the iteration proceeds. The reconstruction error is defined as $\frac{1}{M}\sum_{i=1}^{M}||\mathbf{A}_i-\mathbf{U}\mathbf{B}_i\mathbf{V}^T||_F$. For simplicity, we set $d=10$ for all cases. Six experiments same as those in Section 6.2 are conducted and the results are reported in Fig. 8, where the x-axis denotes the iteration number and the y-axis denotes the error. It can be seen that, after two iterations, B2DPCA converges.

7. Conclusions

A framework of Generalized 2D Principal Component Analysis is proposed to extend the original 2DPCA in three ways: firstly, the essence of 2DPCA is clarified. Secondly, a bilateral 2DPCA scheme is introduced to remove the necessity of more coefficients in representing an image in 2DPCA than in PCA. Thirdly, a kernel-based 2DPCA scheme is introduced to remedy the shortage of 2DPCA in exploring the higher-order statistics among the rows/columns of the input data.

Appendix A. The proof of Theorem 2

Proof. Let $\nabla = \sum_{i=1}^{M} ||\mathbf{A}_i - \mathbf{U}\mathbf{B}_i\mathbf{V}^T||_F^2$. According to the property of trace of matrix, we have

$$\begin{aligned}
\nabla &= \sum_{i=1}^{M} tr((\mathbf{A}_i - \mathbf{U}\mathbf{B}_i\mathbf{V}^T)(\mathbf{A}_i - \mathbf{U}\mathbf{B}_i\mathbf{V}^T)^T) \\
&= \sum_{i=1}^{M} tr(\mathbf{A}_i\mathbf{A}_i^T) + tr(\mathbf{U}\mathbf{B}_i\mathbf{V}^T\mathbf{U}\mathbf{B}_i^T\mathbf{U}^T) - 2tr(\mathbf{U}\mathbf{B}_i\mathbf{V}^T\mathbf{A}_i^T) \\
&= \sum_{i=1}^{M} tr(\mathbf{A}_i\mathbf{A}_i^T) + tr(\mathbf{U}\mathbf{B}_i\mathbf{B}_i^T\mathbf{U}^T) - 2\sum_{i=1}^{M} tr(\mathbf{U}\mathbf{B}_i\mathbf{V}^T\mathbf{A}_i^T) \\
&= \sum_{i=1}^{M} tr(\mathbf{A}_i\mathbf{A}_i^T) + \sum_{i=1}^{M} tr(\mathbf{B}_i^T\mathbf{U}^T\mathbf{U}\mathbf{B}_i) + 2tr(\mathbf{U}\mathbf{B}_i\mathbf{V}^T\mathbf{A}_i^T) \\
&= \sum_{i=1}^{M} \{tr(\mathbf{A}_i\mathbf{A}_i^T) + tr(\mathbf{B}_i^T\mathbf{B}_i) - 2tr(\mathbf{U}\mathbf{B}_i\mathbf{V}^T\mathbf{A}_i^T)\} \\
&= \sum_{i=1}^{M} \{tr(\mathbf{A}_i\mathbf{A}_i^T) + tr(\mathbf{B}_i\mathbf{B}_i^T) - 2tr(\mathbf{U}\mathbf{B}_i\mathbf{V}^T\mathbf{A}_i^T)\}
\end{aligned}$$

where the second term derives from the facts that (1) both $\mathbf{U}$ and $\mathbf{V}$ have orthonormal columns, and (2) $tr(\mathbf{AB}) = tr(\mathbf{BA})$ for any two matrices.

Since the first term is a constant, the minimization of Eq. (8) is equivalent to minimizing:

$$\mathbf{J} = \sum_{i=1}^{M} \{tr(\mathbf{B}_i\mathbf{B}_i^T) - 2tr(\mathbf{U}\mathbf{B}_i\mathbf{V}^T\mathbf{A}_i^T)\} \tag{15}$$

Let,

$$\frac{\partial \mathbf{J}}{\partial \mathbf{B}_i} = 2\sum_{i=1}^{M} \{\mathbf{B}_i - \mathbf{U}^T\mathbf{A}_i\mathbf{V}\} = 0 \tag{16}$$

Therefore, only if $\mathbf{B}_i = \mathbf{U}^T\mathbf{A}_i\mathbf{V}$, the minimum value of $\mathbf{J}$ can be achieved. We substitute $\mathbf{B}_i$ in Eq. (8) by

$$\begin{aligned}
\mathbf{U}^T\mathbf{A}_i\mathbf{V} : \nabla &= \sum_{i=1}^{M} tr((\mathbf{A}_i - \mathbf{U}\mathbf{B}_i\mathbf{V}^T)(\mathbf{A}_i - \mathbf{U}\mathbf{B}_i\mathbf{V}^T)^T) \\
&= \sum_{i=1}^{M} \{tr(\mathbf{A}_i\mathbf{A}_i^T) + tr(\mathbf{B}_i\mathbf{B}_i^T) - 2tr(\mathbf{U}\mathbf{B}_i\mathbf{V}^T\mathbf{A}_i^T)\} \\
&= \sum_{i=1}^{M} \{tr(\mathbf{A}_i\mathbf{A}_i^T) + tr(\mathbf{B}_i\mathbf{B}_i^T) - 2tr(\mathbf{B}_i\mathbf{B}_i^T)\} \\
&= \sum_{i=1}^{M} \{tr(\mathbf{A}_i\mathbf{A}_i^T) - tr(\mathbf{B}_i\mathbf{B}_i^T)\} \\
&= \sum_{i=1}^{M} ||\mathbf{A}_i||_F^2 - \sum_{i=1}^{M} ||\mathbf{B}_i||_F^2 \\
&= \sum_{i=1}^{M} ||\mathbf{A}_i||_F^2 - \sum_{i=1}^{M} ||\mathbf{U}^T\mathbf{A}_i\mathbf{V}||_F^2
\end{aligned}$$

where the first term is a constant, therefore, the minimization of Eq. (8) is equivalent to the maximization of the following Eq. (17) and the solutions that maximize Eq. (17) are the optimal ones.

$$\sigma = \sum_{i=1}^{M} ||\mathbf{U}^T\mathbf{A}_i\mathbf{V}||_F^2 \tag{17}$$

□

Appendix B. The proof of Theorem 3

Proof. From Eqs. (13) and (14), we have $\mathbf{v}_i = (1/\lambda_i)\mathbf{C}^{\Phi}\mathbf{v}_i$.

$$v_i = \frac{1}{\lambda_i}\left[\frac{1}{M}\sum_{k=1}^{M} \hat{\Phi}(A_k)^T\hat{\Phi}(A_k)\right]v_i \tag{18}$$

Another form of $\mathbf{C}^{\Phi}$ is

$$\mathbf{C}^{\Phi} = \frac{1}{M}\Psi^{\Phi}(\Psi^{\Phi})^T \tag{19}$$

where

$$\Psi^{\Phi} = [[\hat{\Phi}(A_1^1)^T, \ldots, \hat{\Phi}(A_1^m)^T], \ldots, [\hat{\Phi}(A_M^1)^T, \ldots, \hat{\Phi}(A_M^m)^T]] \tag{20}$$

From Eqs. (18)–(20), we have,

$$v_i = \frac{1}{\lambda_i M}\Psi^{\Phi}a_i \tag{21}$$

where $\mathbf{a}_i = (\Psi^{\Phi})^T\mathbf{v}_i$ is an $(M \times m)$-dimensional column vector and it is denoted by $\mathbf{a}_i = [\alpha_i^1, \alpha_i^2, \ldots, \alpha_i^{M\times m}]^T$. Thus, the solutions $\mathbf{v}_i$ lie in the span of $\hat{\Phi}(A_k^l)^T$, $k = 1, \ldots, M$; $l = 1, \ldots, m$. That is,

$$v_i = \sum_{k=1}^{M}\sum_{l=1}^{m} \alpha_i^{k\times l}\hat{\Phi}(A_k^l)^T \tag{22}$$

Multiply $\hat{\Phi}(A_g^h)^T$ on both size of Eq. (14), we can get,

$$\lambda_i(\hat{\Phi}(A_g^h)^T \bullet v_i) = (\hat{\Phi}(A_g^h)^T \bullet C^{\Phi}v_i) \tag{23}$$

That is,

$$\lambda_i \sum_{k=1}^{M}\sum_{l=1}^{m} \alpha_i^{k\times l}\left(\hat{\Phi}(\mathbf{A}_g^h)^T\hat{\Phi}(\mathbf{A}_k^l)^T\right)$$
$$= \left(\hat{\Phi}(\mathbf{A}_g^h)^T\left[\frac{1}{M}\sum_{t=1}^{M}\hat{\Phi}(\mathbf{A}_t)^T\hat{\Phi}(\mathbf{A}_t)\sum_{k=1}^{M}\sum_{l=1}^{m}\alpha_i^{k\times l}\hat{\Phi}(\mathbf{A}_k^l)^T\right]\right)$$
$$= \left(\hat{\Phi}(\mathbf{A}_g^h)^T\left[\frac{1}{M}\sum_{p=1}^{M}\sum_{q=1}^{m}\hat{\Phi}(\mathbf{A}_p^q)^T\hat{\Phi}(\mathbf{A}_p^q)\right.\right.$$
$$\left.\left.\times\sum_{k=1}^{M}\sum_{l=1}^{m}\alpha_i^{k\times l}\hat{\Phi}(A_k^l)^T\right]\right)$$
$$= \frac{1}{M}\sum_{k=1}^{M}\sum_{l=1}^{m}\alpha_i^{k\times l}(\hat{\Phi}(A_g^h))^T\sum_{p=1}^{M}\sum_{q=1}^{m}\hat{\Phi}(\mathbf{A}_p^q)^T$$
$$\times\left(\hat{\Phi}(\mathbf{A}_p^q)^T\hat{\Phi}(\mathbf{A}_k^l)^T\right).$$

Defining an $(M\times m)\times(M\times m)$ matrix $\mathbf{K}$ by

$$K_{(k\times l,p\times q)} = \left(\hat{\Phi}(A_k^l)^T\hat{\Phi}(A_p^q)^T\right)$$

The above equation can be converted into:

$$M\lambda_i\mathbf{K}\mathbf{a}_i = \mathbf{K}^2\mathbf{a}_i \tag{24}$$

or

$$M\lambda_i\mathbf{a}_i = \mathbf{K}\mathbf{a}_i \tag{25}$$

Since K is positive semidefinite, K's eigenvalues will be nonnegative, the eigenvalues $\lambda_1\leq\lambda_2,\leq,\ldots,\leq\lambda_{\mathrm{M}\times\mathrm{m}}$ and the corresponding eigenvectors $\mathbf{a}_1$, $\mathbf{a}_2,\ldots,\mathbf{a}_{M\times m}$ can be solved by diagonalizing $\mathbf{K}$, with $\mathbf{a}_{\mathrm{p}},\mathbf{a}_{\mathrm{p}+1},\ldots,\mathbf{a}_{\mathrm{M}\times\mathrm{m}}$ by enforcing the unitilization of the corresponding $\mathbf{v}$ in F, i.e. $(\mathbf{v}_d\cdot\mathbf{v}_d)=1$ for all $d=p,\ldots,M\times m$. In terms of $\mathbf{v}_i=\sum_{k=1}^{M}\sum_{l=1}^{m}\alpha_i^{k\times l}\hat{\Phi}(\mathbf{A}_k^l)^T$, this turns into:

$$1 = \left(\left(\sum_{k=1}^{M}\sum_{l=1}^{m}\alpha_d^{k\times l}\hat{\Phi}(\mathbf{A}_k^l)^T\right)\left(\sum_{p=1}^{M}\sum_{q=1}^{m}\alpha_d^{p\times q}\hat{\Phi}(\mathbf{A}_p^q)^T\right)\right)$$
$$= (\mathbf{a}_d\mathbf{K}\mathbf{a}_d) = \lambda_d(\mathbf{a}_d\mathbf{a}_d).$$

To extract the principal component of each row, we need to project each $\hat{\Phi}(\mathbf{A}_i^j)$ onto the eigenvectors $\mathbf{v}_k$ in $\mathbf{F}$, i.e.

$$\left(\mathbf{v}_k\hat{\Phi}(\mathbf{A}_i^j)\right) = \sum_{p=1}^{M}\sum_{q=1}^{m}\alpha_d^{p\times q}\left(\hat{\Phi}(\mathbf{A}_p^q)^T\hat{\Phi}(\mathbf{A}_i^j)\right).$$

Hence, K2DPCA performed on 2D images can be regarded as KPCA performed on the rows of all the training images. □

References

Belhumeur, P. N., Hespanha, J., & Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(7), 711–720.

Bischof, H., Wildenauer, H., & Leonardis, A. (2004). Illumination insensitive recognition using eigenimages. *Computer Vision and Image Understanding*, *95*, 86–104.

Cevikalp, H., Neamtu, M., Wilkes, M., & Barkana, A. (2005). Discriminative common vectors for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(1), 4–13.

Epstein, R., Hallinan, P., & Yuille, A. L. (1995). Eigenimages suffice: An empirical investigation of low-dimensional lighting models. *IEEE workshop on physics-based modeling in computer vision* pp. 108–116.

Fukunnaga, K. (1991). Introduction to statistical pattern recognition. Academic Press (pp.38–40).

Hallinan, P., et al. (1994). A low-dimensional representation of human faces for arbitrary lighting conditions. *IEEE conference on computer vision and pattern recognition, Seattle, WA* pp. 995–999.

Kirby, M., & Sirovich, L. (1990). Application of the KL procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(1), 103–108.

Kong, H., Li, X., Wang, L., Teoh, E. K., Wang, J. G., & Venkateswarlu, R. (2005). *Generalized 2D principal component analysis International joint conference on neural networks, Montréal, Canada.*

Liu, Q., Huang, R., Lu, H., & Ma, S. (2002). Face recognition using Kernel based fisher discriminant analysis. *IEEE international conference on face and gesture recognition, Washington, DC.*

Liu, X., Chen, T., & Bhagavatula, V. (2003). Face authentication for multiple subjects using eigenflow. *Pattern Recognition, Special Issue on Biometric*, *36*(2), 313–328.

Lu, J., Plataniotis, K. N., & Venetsanopoulos, A. N. (2003). Face recognition using kernel direct discriminant analysis algorithms. *IEEE Transactions on Neural Networks*, *14*(1), 117–126.

Murase, H., & Nayar, S. (1995). Visual learning and recognition of 3d objects from appearance. *International Journal on Computer Vision*, *14*(1), 5–24.

Pentland, A., Moghaddam, B., & Starner, T. (1994). View-based and modular eigenspaces for face recognition. *IEEE conference on computer vision and pattern recognition, Seattle, WA* (Seattle WA).

Ramamoorthi, R. (2002). Analytic PCA construction for theoretical analysis of lighting variability in images of a lambertian object. *IEEE transactions on pattern analysis and machine intelligence* pp. 1322–1333.

Scholkopf, B., Smola, A., & Muller, K. R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* , 1299–1319.

Shashua, A. (1992). Geometry and photometry in 3D visual recognition. PhD Thesis, MIT.

Sirovich, L., & Kirby, M. (1987). Low-dimensional procedure for the characterization of human faces. *Journal of Optical Society of America*, *4*(3), 519–524.

Swets, D. L., & Weng, J. (1996). Using discriminant eigenfeatures for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*(8), 831–836.

Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, *3*(1), 71–86.

Yang, M. H. (2002). Kernel Eigenface vs. Kernel Fisherface: Face recognition using Kernel methods. *IEEE international conference on face and gesture recognition, Washington, DC.*

Yang, J., Zhang, D., Frangi, A. F., & Yang, J. (2004). 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.

Zhao, W. (2000). Discriminant component analysis for face recognition. *International Conference on Pattern Recognition* .

Zhao, L., & Yang, Y. (1999). Theoretical analysis of illumination in PCA-based vision systems. *Pattern Recognition*, *32*(4), 547–564.