Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition

Jian Yang, David Zhang, *Senior Member*, *IEEE*, Alejandro F. Frangi, and Jing-yu Yang

Abstract—In this paper, a new technique coined two-dimensional principal component analysis (2DPCA) is developed for image representation. As opposed to PCA, 2DPCA is based on 2D image matrices rather than 1D vectors so the image matrix does not need to be transformed into a vector prior to feature extraction. Instead, an *image covariance matrix* is constructed directly using the original image matrices, and its eigenvectors are derived for image feature extraction. To test 2DPCA and evaluate its performance, a series of experiments were performed on three face image databases: ORL, AR, and Yale face databases. The recognition rate across all trials was higher using 2DPCA than PCA. The experimental results also indicated that the extraction of image features is computationally more efficient using 2DPCA than PCA.

Index Terms—Principal Component Analysis (PCA), Eigenfaces, feature extraction, image representation, face recognition.

1 Introduction

PRINCIPAL component analysis (PCA), also known as Karhunen-Loeve expansion, is a classical feature extraction and data representation technique widely used in the areas of pattern recognition and computer vision. Sirovich and Kirby [1], [2] first used PCA to efficiently represent pictures of human faces. They argued that any face image could be reconstructed approximately as a weighted sum of a small collection of images that define a facial basis (eigenimages), and a mean image of the face. Within this context, Turk and Pentland [3] presented the well-known Eigenfaces method for face recognition in 1991. Since then, PCA has been widely investigated and has become one of the most successful approaches in face recognition [4], [5], [6], [7]. Penev and Sirovich [8] discussed the problem of the dimensionality of the "face space"when eigenfaces are used for representation. Zhao and Yang [9] tried to account for the arbitrary effects of illumination in PCA-based vision systems by generating an analytically closed-form formula of the covariance matrix for the case with a special lighting condition and then generalizing to an arbitrary illumination via an illumination equation. However, Wiskott et al. [10] pointed out that PCA could not capture even the simplest invariance unless this information is explicitly provided in the training data. They proposed a technique known as elastic bunch graph matching to overcome the weaknesses of PCA.

Recently, two PCA-related methods, *independent component analysis* (ICA) and *kernel principal component analysis* (Kernel PCA) have been of wide concern. Bartlett et al. [11] and Draper et al. [12] proposed using ICA for face representation and found that it was better than PCA when cosines were used as the similarity measure (however, their performance was not significantly different if the Euclidean distance is used). Yang [14] used Kernel PCA for face feature extraction and recognition and showed that the Kernel Eigenfaces method outperforms the classical Eigenfaces method. However, ICA and Kernel PCA are both computationally more expensive than PCA. The experimental results in [14] showed the ratio of the computation time required by ICA, Kernel PCA, and PCA is, on average, 8.7: 3.2: 1.0.

In the PCA-based face recognition technique, the 2D face image matrices must be previously transformed into 1D image vectors. The resulting image vectors of faces usually lead to a high-dimensional image vector space, where it is difficult to evaluate the covariance matrix accurately due to its large size and the relatively small number of training samples. Fortunately, the eigenvectors (eigenfaces) can be calculated efficiently using the SVD techniques [1], [2] and the process of generating the covariance matrix is actually avoided. However, this does not imply that the eigenvectors can be evaluated accurately in this way since the eigenvectors are statistically determined by the covariance matrix, no matter what method is adopted for obtaining them.

In this paper, a straightforward image projection technique, called *two-dimensional principal component analysis* (2DPCA), is developed for image feature extraction. As opposed to conventional PCA, 2DPCA is based on 2D matrices rather than 1D vectors. That is, the image matrix does not need to be previously transformed into a vector. Instead, an *image covariance matrix* can be constructed directly using the original image matrices. In contrast to the covariance matrix of PCA, the size of the *image covariance matrix* using 2DPCA is much smaller. As a result, 2DPCA has two important advantages over PCA. First, it is easier to evaluate the covariance matrix accurately. Second, less time is required to determine the corresponding eigenvectors.

The remainder of this paper is organized as follows: In Section 2, the idea of the proposed 2DPCA method and its algorithm are described. The image reconstruction method using 2DPCA is developed in Section 3. In Section 4, experimental results are presented for the ORL, the AR, and the Yale face image databases to demonstrate the effectiveness and robustness of 2DPCA. Finally, conclusions are presented in Section 5.

2 Two-Dimensional Principal Component Analysis

2.1 Idea and Algorithm

Let $\mathbf{X}$ denote an n-dimensional unitary column vector. Our idea is to project image $\mathbf{A}$, an $m \times n$ random matrix, onto $\mathbf{X}$ by the following linear transformation [15], [19]:

$$\mathbf{Y} = \mathbf{A}\mathbf{X}. \tag{1}$$

Thus, we obtain an m-dimensional projected vector $\mathbf{Y}$, which is called the projected feature vector of image $\mathbf{A}$. How do we determine a good projection vector $\mathbf{X}$? In fact, the total scatter of the projected samples can be introduced to measure the discriminatory power of the projection vector $\mathbf{X}$. The total scatter of the projected samples can be characterized by the trace of the covariance matrix of the projected feature vectors. From this point of view, we adopt the following criterion:

$$J(\mathbf{X}) = \mathrm{tr}(\mathbf{S}_x), \tag{2}$$

- *J. Yang is with the Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong, Computer Vision Group, Aragon Institute of Engineering Research, Universidad de Zaragoza, E-50018 Zaragoza, Spain, and the Department of Computer Science, Nanjing University of Science and Technology, Nanjing 210094, P.R. China. E-mail: jyang@unizar.es.*
- *D. Zhang is with the Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong. E-mail: csdzhang@comp.polyu.edu.hk.*
- *A.F. Frangi is with the Computer Vision Group, Aragon Institute of Engineering Research, Universidad de Zaragoza, E-50018 Zaragoza, Spain. E-mail: afrangi@unizar.es.*
- *J.-y. Yang is with the Department of Computer Science, Nanjing University of Science and Technology, Nanjing 210094, P.R. China. E-mail: yangjy@mail.njust.edu.cn.*

Manuscript received 10 Sept. 2002; revised 4 Mar. 2003; accepted 17 Mar. 2003.
Recommended for acceptance by Y. Amit.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 117299.

Fig. 1. Five sample images of one subject in the ORL face database.

where $\mathbf{S}_x$ denotes the covariance matrix of the projected feature vectors of the training samples and $\mathrm{tr}(\mathbf{S}_x)$ denotes the trace of $\mathbf{S}_x$. The physical significance of maximizing the criterion in (2) is to find a projection direction $\mathbf{X}$, onto which all samples are projected, so that the total scatter of the resulting projected samples is maximized. The covariance matrix $\mathbf{S}_x$ can be denoted by

$$\begin{aligned}\mathbf{S}_x &= E(\mathbf{Y}-E\mathbf{Y})(\mathbf{Y}-E\mathbf{Y})^T = E[\mathbf{AX}-E(\mathbf{AX})][\mathbf{AX}-E(\mathbf{AX})]^T \\ &= E[(\mathbf{A}-E\mathbf{A})\mathbf{X}][(\mathbf{A}-E\mathbf{A})\mathbf{X}]^T.\end{aligned}$$

So,

$$\mathrm{tr}(\mathbf{S}_x) = \mathbf{X}^T[E(\mathbf{A}-E\mathbf{A})^T(\mathbf{A}-E\mathbf{A})]\mathbf{X}. \tag{3}$$

Let us define the following matrix

$$\mathbf{G}_t = E[(\mathbf{A}-E\mathbf{A})^T(\mathbf{A}-E\mathbf{A})]. \tag{4}$$

The matrix $\mathbf{G}_t$ is called the *image covariance* (*scatter*) *matrix*. It is easy to verify that $\mathbf{G}_t$ is an $n \times n$ nonnegative definite matrix from its definition. We can evaluate $\mathbf{G}_t$ directly using the training image samples. Suppose that there are M training image samples in total, the jth training image is denoted by an $m \times n$ matrix $\mathbf{A}_j (j = 1, 2, \cdots, M)$, and the average image of all training samples is denoted by $\bar{\mathbf{A}}$. Then, $\mathbf{G}_t$ can be evaluated by

$$\mathbf{G}_t = \frac{1}{M}\sum_{j=1}^{M}(\mathbf{A}_j-\bar{\mathbf{A}})^T(\mathbf{A}_j-\bar{\mathbf{A}}). \tag{5}$$

Alternatively, the criterion in (2) can be expressed by

$$J(\mathbf{X}) = \mathbf{X}^T\mathbf{G}_t\mathbf{X}, \tag{6}$$

where $\mathbf{X}$ is a unitary column vector. This criterion is called the *generalized total scatter criterion*. The unitary vector $\mathbf{X}$ that maximizes the criterion is called the optimal projection axis. Intuitively, this means that the total scatter of the projected samples is maximized after the projection of an image matrix onto $\mathbf{X}$.

The optimal projection axis $\mathbf{X}_{opt}$ is the unitary vector that maximizes $J(\mathbf{X})$, i.e., the eigenvector of $\mathbf{G}_t$ corresponding to the largest eigenvalue [19]. In general, it is not enough to have only one optimal projection axis. We usually need to select a set of projection axes, $\mathbf{X}_1, \cdots, \mathbf{X}_d$, subject to the orthonormal constraints and maximizing the criterion $J(\mathbf{X})$, that is,

$$\begin{cases}\{\mathbf{X}_1, \cdots, \mathbf{X}_d\} = \arg\max J(\mathbf{X}) \\ \mathbf{X}_i^T\mathbf{X}_j = 0, i \neq j, i, j = 1, \cdots, d.\end{cases} \tag{7}$$

In fact, the optimal projection axes, $\mathbf{X}_1, \cdots, \mathbf{X}_d$, are the orthonormal eigenvectors of $\mathbf{G}_t$ corresponding to the first d largest eigenvalues.

2.2 Feature Extraction

The optimal projection vectors of 2DPCA, $\mathbf{X}_1, \cdots, \mathbf{X}_d$, are used for feature extraction. For a given image sample $\mathbf{A}$, let

$$\mathbf{Y}_k = \mathbf{AX}_k, k = 1, 2, \cdots, d. \tag{8}$$

Then, we obtain a family of projected feature vectors, $\mathbf{Y}_1, \cdots, \mathbf{Y}_d$, which are called the *principal component* (*vectors*) of the sample image $\mathbf{A}$. It should be noted that each *principal component* of 2DPCA is a vector, whereas the *principal component* of PCA is a scalar.

The principal component vectors obtained are used to form an $m \times d$ matrix $\mathbf{B} = [\mathbf{Y}_1, \cdots, \mathbf{Y}_d]$, which is called the *feature matrix* or *feature image* of the image sample $\mathbf{A}$.

2.3 Classification Method

After a transformation by 2DPCA, a feature matrix is obtained for each image. Then, a nearest neighbor classifier is used for classification. Here, the distance between two arbitrary feature matrices, $\mathbf{B}_i = [\mathbf{Y}_1^{(i)}, \mathbf{Y}_2^{(i)}, \cdots, \mathbf{Y}_d^{(i)}]$ and $\mathbf{B}_j = [\mathbf{Y}_1^{(j)}, \mathbf{Y}_2^{(j)}, \cdots, \mathbf{Y}_d^{(j)}]$, is defined by

$$d(\mathbf{B}_i, \mathbf{B}_j) = \sum_{k=1}^{d}\left\|\mathbf{Y}_k^{(i)} - \mathbf{Y}_k^{(j)}\right\|_2, \tag{9}$$

where $\|\mathbf{Y}_k^{(i)} - \mathbf{Y}_k^{(j)}\|_2$ denotes the Euclidean distance between the two principal component vectors $\mathbf{Y}_k^{(i)}$ and $\mathbf{Y}_k^{(j)}$.

Suppose that the training samples are $\mathbf{B}_1, \mathbf{B}_2, \cdots, \mathbf{B}_M$ (where M is the total number of training samples), and that each of these samples is assigned a given identity (class) ω_k. Given a test sample $\mathbf{B}$, if $d(\mathbf{B}, \mathbf{B}_l) = \min_j d(\mathbf{B}, \mathbf{B}_j)$ and $\mathbf{B}_l \in \omega_k$, then the resulting decision is $\mathbf{B} \in \omega_k$.

3 2DPCA-Based Image Reconstruction

In the Eigenfaces method, the principal components and eigenvectors (eigenfaces) can be combined to reconstruct the image of a face. Similarly, 2DPCA can be used to reconstruct a face image in the following way.

Suppose the orthonormal eigenvectors corresponding to the first d largest eigenvectors of the image covariance matrix $\mathbf{G}_t$ are $\mathbf{X}_1, \cdots, \mathbf{X}_d$. After the image samples are projected onto these axes, the resulting principal component vectors are $\mathbf{Y}_k = \mathbf{AX}_k$ $(k = 1, 2, \cdots, d)$. Let $\mathbf{V} = [\mathbf{Y}_1, \cdots, \mathbf{Y}_d]$ and $\mathbf{U} = [\mathbf{X}_1, \cdots, \mathbf{X}_d]$, then

$$\mathbf{V} = \mathbf{AU}. \tag{10}$$

Since $\mathbf{X}_1, \cdots, \mathbf{X}_d$ are orthonormal, from (10), it is easy to obtain the reconstructed image of sample $\mathbf{A}$:

$$\tilde{\mathbf{A}} = \mathbf{VU}^T = \sum_{k=1}^{d}\mathbf{Y}_k\mathbf{X}_k^T. \tag{11}$$

Let $\tilde{\mathbf{A}}_k = \mathbf{Y}_k\mathbf{X}_k^T (k = 1, 2, \cdots, d)$, which is of the same size as image $\mathbf{A}$, and represents the *reconstructed subimage* of $\mathbf{A}$. That is, image $\mathbf{A}$ can be approximately reconstructed by adding up the first d subimages. In particular, when the selected number of principal component vectors $d = n$ (n is the total number of eigenvectors of $\mathbf{G}_t$), we have $\tilde{\mathbf{A}} = \mathbf{A}$, i.e., the image is completely reconstructed by its principal component vectors without any loss of information. Otherwise, if $d < n$, the reconstructed image $\tilde{\mathbf{A}}$ is an approximation for $\mathbf{A}$.

4 Experiments and Analysis

The proposed 2DPCA method was used for face recognition and tested on three well-known face image databases (ORL, AR, and Yale). The ORL database was used to evaluate the performance of 2DPCA under conditions where the pose and sample size are varied. The AR database was employed to test the performance of the system under conditions where there is a variation over time, in facial expressions, and in lighting conditions. The Yale database was used to examine the system performance when both facial expressions and illumination are varied.

4.1 Experiments on the ORL Database

The ORL database (http://www.cam-orl.co.uk) contains images from 40 individuals, each providing 10 different images. For some subjects, the images were taken at different times. The facial

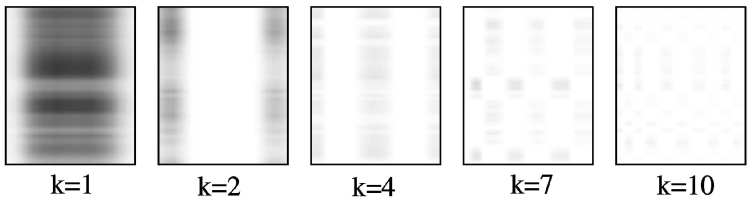

Fig. 2. Some reconstructed subimages are shown in inverse color.

expressions (open or closed eyes, smiling or nonsmiling) and facial details (glasses or no glasses) also vary. The images were taken with a tolerance for some tilting and rotation of the face of up to 20 degrees. Moreover, there is also some variation in the scale of up to about 10 percent. All images are grayscale and normalized to a resolution of 92×112 pixels. Five sample images of one person from the ORL database are shown in Fig. 1.

First, an experiment was performed using the first five image samples per class for training, and the remaining images for test. Thus, the total number of training samples and testing samples were both 200. The 2DPCA algorithm was first used for feature extraction. Here, the size of image covariance matrix $\mathbf{G}_t$ was 92×92, so it was very easy to calculate its eigenvectors. We chose the eigenvectors corresponding to 10 largest eigenvalues, $\mathbf{X}_1, \cdots, \mathbf{X}_{10}$, as projection axes. After the projection of the image sample onto these axes using (8), we obtained ten principal component vectors, $\mathbf{Y}_1, \cdots, \mathbf{Y}_{10}$. Taking the last image in Fig. 1 as an example, we can determine its 10 constructed subimages, $\tilde{\mathbf{A}}_k = \mathbf{Y}_k \mathbf{X}_k^T,\ k = 1, 2, \cdots, 10$. Some of these subimages are shown in Fig. 2 in reverse color for the sake of clarity. Moreover, the magnitude of $\mathbf{G}_t$'s eigenvalues is plotted in decreasing order in Fig. 3.

As observed in Fig. 2, the first subimage contains most of the energy of the original image. The other ones show the detailed local information from different levels. As the value of k increases, the information (the energy of image) contained in $\tilde{\mathbf{A}}_k$ becomes gradually weaker. Fig. 3 shows the magnitude of the eigenvalues quickly converges to zero, which is exactly consistent with the results of Fig. 2. Thus, we can conclude that the energy of an image is concentrated on its first small number of component vectors. Therefore, it is reasonable to use these component vectors to represent the image for recognition purposes.

On the other hand, by adding up the first d subimages together, we obtain an approximate reconstruction of the original image. Fig. 4 shows five reconstructed images of the last image in Fig. 1 by adding the first d (d = 2, 4, 6, 8, 10) subimages together. The reconstructed images become clearer as the number of subimages is increased. For comparison, the PCA (Eigenfaces) was also used to represent and reconstruct the same face image. Fig. 4 also shows the reconstructed images as the number of principal components d is set to 5, 10, 20, 30, and 40. The PCA did not perform as well in the reconstruction of this image.

Now, let us design a series of experiments to compare the performance of 2DPCA and PCA (Eigenfaces) under conditions where the sample size is varied. Here, five tests were performed with a varying number of training samples. More specifically, in the kth test, we used the first k image samples per class for training and the remaining samples for testing. The proposed 2DPCA method and the PCA (Eigenfaces) method were used for feature extraction. Finally, a nearest neighbor classifier was employed for classification. Note that in 2DPCA, (9) is used to calculate the distance between two feature matrices (formed by the principal component vectors). In PCA (Eigenfaces), the common Euclidean distance measure is adopted. Table 1 presents the top recognition accuracy of PCA and 2DPCA, which corresponds to different numbers of training samples. The performance of 2DPCA is better than PCA. Here, it should be pointed out that PCA used all components (at most $M - 1$, where M is the total number of training samples) for achieving the maximal recognition accuracy when there are one or two samples per person for training.

The 2DPCA method is also superior to PCA in terms of computational efficiency for feature extraction. Table 2 indicates that feature extraction by 2DPCA takes much less time. As the number of training samples per class is increased, the relative gain between 2DPCA and PCA becomes more apparent.

However, one disadvantage of 2DPCA (compared to PCA) is that more coefficients are needed to represent an image. From Table 1, it is clear that dimension of the 2DPCA feature vector ($112 \times d$) is always much higher than PCA at top recognition accuracy. How do we reduce the dimension of 2DPCA? A simple strategy is to use PCA for further dimensional reduction after 2DPCA, i.e., *2DPCA plus PCA*. To test this strategy, we derive eight component vectors (112×8 features in total) using 2DPCA when there are five samples per class for training. Then, PCA is used for the second feature extraction and a nearest neighbor classifier is employed. The classification results are shown in Fig. 5. This figure indicates that the performance of *2DPCA plus PCA* is still better than that of PCA only for the same dimensionality.

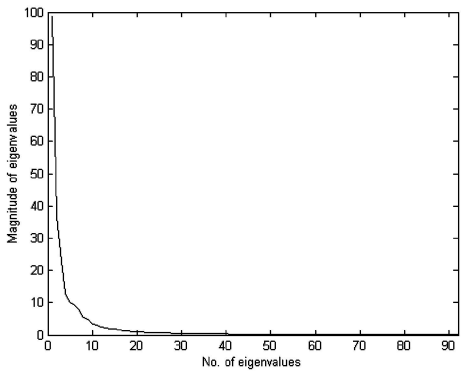

Fig. 3. The plot of the magnitude of the eigenvalues in decreasing order.

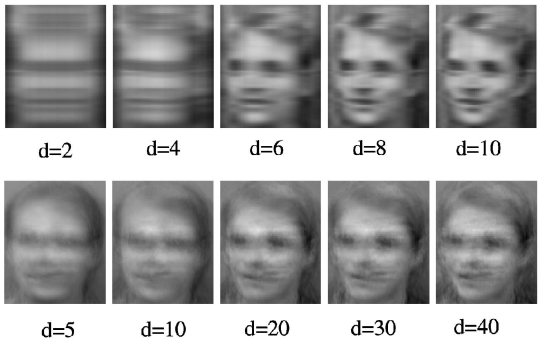

Fig. 4. Some reconstructed images based on 2DPCA (upper) and PCA (lower).

TABLE 1
Comparison of the Top Recognition Accuracy (%) of PCA versus 2DPCA

# Training samples / class	1 *	2 *	3	4 *	5
PCA (Eigenfaces)	66.9 (39)	84.7 (79)	88.2 (95)	90.8 (60)	93.5 (37)
2DPCA	76.7 (112×2)	89.1 (112×2)	91.8 (112×6)	95.0 (112×5)	96.0 (112×3)

The values in parentheses denote the dimension of feature vectors for the best recognition accuracy. Note that the best choices of the number of the components for the top recognition accuracy depend on the test data and are not known beforehand in a real problem. The asterisks indicate a statistically significant difference between PCA and 2DPCA at a significance level of 0.05 in the trials.

TABLE 2
Comparison of CPU Time (s) for Feature Extraction Using the ORL (CPU: Pentium III 800MHz, RAM: 256 Mb)

# Training samples / class	1	2	3	4	5
PCA (Eigenfaces)	44.45	89.00	139.36	198.95	304.61
2DPCA	10.76	11.23	12.59	13.40	14.03

The performance of 2DPCA was also compared with other methods, including Fisherfaces [16], ICA [13], [14], and Kernel Eigenfaces [14]. In these comparisons, two experimental strategies were adopted. One strategy was "using the first five images per class for training,"which is mentioned above. The other was the *leave-one-out* strategy, that is, the image of one person is removed from the data set and all of the remaining images are used for training. The experimental results under both strategies are listed in Table 3. 2DPCA was better than other methods except for the recognition rate compared to Fisherfaces in the "leave-one-out"strategy.

4.2 Experiment on the AR Database

The AR face database [17], [18] contains over 4,000 color face images of 126 people (70 men and 56 women), including frontal views of faces with different facial expressions, lighting conditions and occlusions. The pictures of most persons were taken in two sessions (separated by two weeks). Each section contains 13 color images and 120 individuals (65 men and 55 women) participated in both sessions. The images of these 120 individuals were selected and used in our experiment. Only the full facial images were considered here (no attempt was made to handle occluded face recognition in each session). We manually cropped the face portion of the image and then normalized it to 50×40 pixels. The normalized images of one person are shown in Fig. 6, where Figs. 6a, 6b, 6c, 6d, 6e, 6e, 6f, and 6g are from Session 1, and Figs. 6n, 6o, 6p, 6q, 6r, 6s, and 6t are from Session 2. The details of the images are: Fig. 6a neutral expression, Fig. 6b smile, Fig. 6c anger, Fig. 6d scream, Fig. 6e left light on; Fig. 6f right light on; Fig. 6g all sides light on; and Figs. 6n, 6o, 6p, 6q, 6r, 6s, and 6t were taken under the same conditions as Figs. 6a, 6b, 6c, 6d, 6e, 6e, 6f, and 6g.

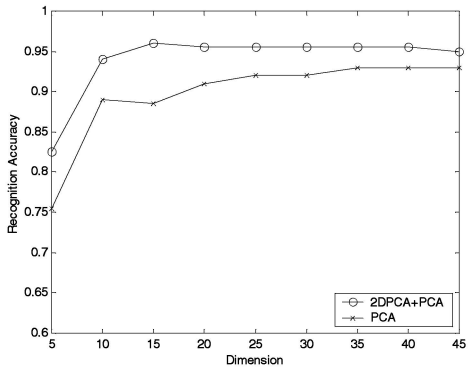

Fig. 5. Comparison of 2DPCA plus PCA and PCA alone on the ORL database.

4.2.1 Variations Over Time

In this experiment, images from the first session (i.e., Figs. 6a, 6b, 6c, 6d, 6e, 6e, 6f, and 6g) were used for training, and images from the second session (i.e., Figs. 6n, 6o, 6p, 6q, 6r, 6s, and 6t) were used for testing. Thus, the total number of training samples was 840. Since the two sessions were separated by an interval of two weeks, the aim of this experiment was to compare the performance of PCA and 2DPCA under the conditions where there are changes over time.

Features were extracted using PCA and 2DPCA, respectively. Then, 100 PCA component features were obtained and 10 2DPCA component feature vectors. The number of selected 2DPCA component feature vectors varied from 1 to 10. The number of selected PCA component features varied from 10 to 100 in intervals of 10. Based on the selected features, a nearest neighbor classifier was employed for classification. The corresponding recognition accuracies are illustrated in Fig. 7. In general, 2DPCA performed better than PCA. The top recognition accuracy of 2DPCA was 67.6 percent using 10 feature vectors, but 66.2 percent using PCA with 100 component features.

Feature extraction times for both methods are summarized in Table 4. Feature extraction with 2DPCA is more than 20 times faster than PCA, mainly because the latter involves calculating the eigenvectors of an 840×840 matrix, whereas 2DPCA calculates the eigenvectors of a 40×40 matrix.

4.2.2 Variations in Facial Expressions

In this experiment, the objective was to compare PCA and 2DPCA under varying facial expressions. We selected images Figs. 6a, 6b, 6c, and 6d and Figs. 6n, 6o, 6p, and 6q), which involve variations in facial expressions. Figs. 6a and 6n were used for training and the others (i.e., Figs. 6b and 6b, 6c, and 6d and Figs. 6o, 6p, and 6q) were used for testing. Thus, the total number of training samples is 240.

As in the previous experiment, PCA was used to extract 100 principal component features and 2DPCA to extract 10 principal component feature vectors. Fig. 7 shows the recognition accuracy under a varying number of selected features (or feature vectors). The top recognition accuracy and the time consumed for feature extraction are listed in Table 4. Again, 2DPCA was more effective and efficient than PCA.

4.2.3 Variations in Lighting Conditions

In this experiment, our aim was to compare PCA and 2DPCA under varying illumination. Images with varying lighting conditions were selected first. The selected sample set included Figs. 6a, 6e, 6f, and 6g from the first session and Figs. 6n, 6r, 6s, and 6t from the second session. From this set, we arbitrarily chose two samples for training, one from the first session and another from the second. The remaining samples were used for testing. Thus, there

TABLE 3
Comparison of 2DPCA with Other Methods Using the ORL Database

Strategy	Method	Recognition rate
Using the first five images for training	Fisherfaces	94.5%
	ICA [13] *	85.0%
	Kernel Eigenfaces	94.0%
	2DPCA	96.0%
Leave-one-out	Fisherfaces [14]	98.5%
	ICA [14] *	93.8 %
	Eigenfaces [14]	97.5%
	Kernel Eigenfaces [14]	98.0%
	2DPCA	98.3%

Note that ICA is tested using Euclidean distance in [14]. Note that the asterisks indicate a statistically significant difference between the marked method and 2DPCA at a significance level of 0.05.

Fig. 6. Sample images for one subject of the AR database.

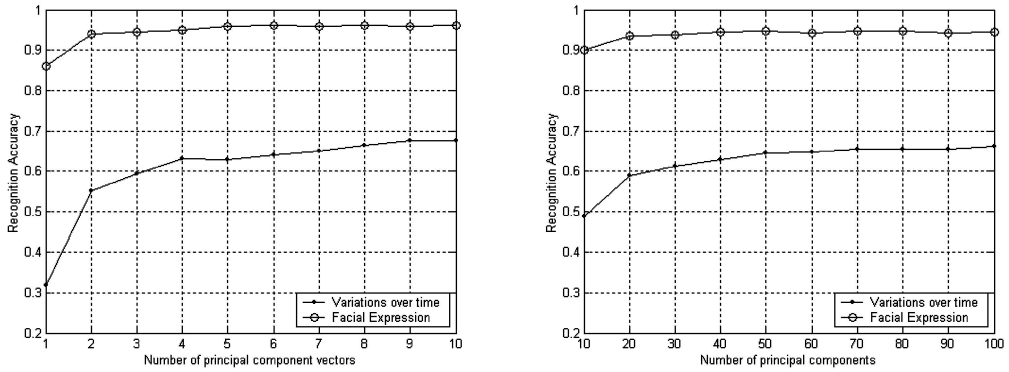

Fig. 7. Performance of 2DPCA and PCA under the condition of variations over time and in facial expressions.

were 16 possible sets of training samples. Based on these sets of training samples, we performed 16 tests. In each test, the performance of PCA and 2DPCA was compared. The experimental results of training sample sets {(a), (n)}, {(e), (s)}, and {(f), (t)} are shown in Fig. 8, with the recognition accuracy of PCA and 2DPCA with varying number of selected features. Fig. 9 illustrates the top recognition accuracy of PCA and 2DPCA from each test. This figure indicates that the performance of 2DPCA is much better than PCA under conditions where lighting is varied. Table 4 shows the average recognition accuracy from the 16 tests. The average recognition accuracy of 2DPCA was 89.8 percent, more than 10 percent higher than PCA. Table 4 also indicates that feature extraction using 2DPCA was much faster than PCA.

4.3 Experiment on the Yale Database

The last experiment was performed using the Yale face database, which contains 165 images of 15 individuals (each person has 11 different images) under various facial expressions and lighting conditions. Each image was manually cropped and resized to 100×80 pixels in this experiment.

In this experiment, the *leave-one-out* strategy was adopted. The experimental results using 2DPCA, PCA (Eigenfaces), ICA, and Kernel Eigenfaces are listed in Table 5. The recognition rate of 2DPCA was superior to PCA, ICA and Kernel Eigenfaces.

4.4 Evaluation of the Experimental Results

The above experiments showed that the recognition rate of 2DPCA is always higher than PCA. But, is this difference statistically

TABLE 4
Comparison of PCA and 2DPCA Using the AR Database under the Condition of Variations over Time, in Facial Expression and Illumination

Experiment		Recognition accuracy (%)	Time of feature extraction (s)	Size of the matrix
Variations over time	PCA	66.2	434.87	840×840
	2DPCA	67.6	16.26	40×40
Facial expression	PCA	94.7	130.42	240×240
	2DPCA	96.1	7.25	40×40
Illumination * (mean)	PCA	78.0	129.56	240×240
	2DPCA	89.8	8.32	40×40

Note that the asterisk indicates a statistically significant difference between PCA and 2DPCA at a significance level of 0.05 in the trial.

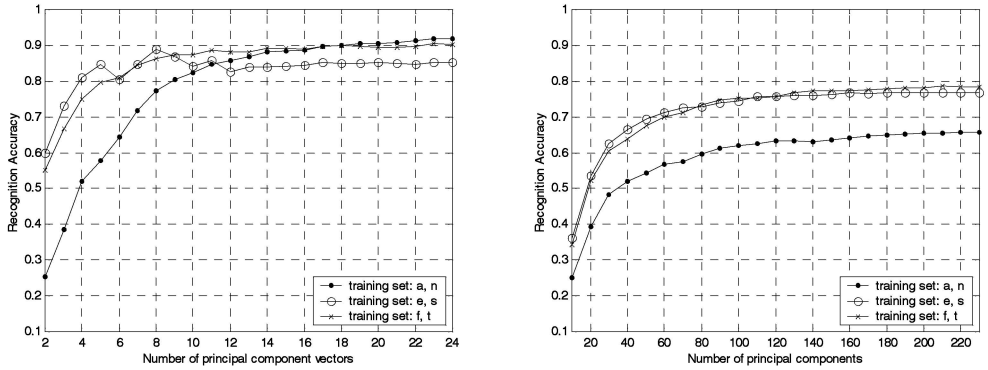

Fig. 8. Recognition accuracy of 2DPCA and PCA for three different training sample sets: {a, n}, {e, s}, and {f, t} under the condition of variations in illumination.

significant? In this section, we evaluate the experimental results using the null hypothesis statistical test based on Bernoulli model [20], [21]. If the resulting p-value is below the desired significance level (i.e., 0.05), the null hypothesis is rejected and the performance difference between two algorithms are considered statistically significant. The evaluation results based on the statistical test (1-tailed) were noted in Tables 1, 3, 4, and 5 and summarized as follows:

1. For the ORL database, 2DPCA outperformed PCA significantly in the trials with 1, 2, and 4 training samples per class ($p = 0.0017$, 0.0492, and 0.0361, respectively).
2. For the AR database, 2DPCA outperformed PCA significantly under condition of variations in illuminations ($p < 0.001$).
3. For the Yale database, 2DPCA was significantly better than PCA and the others ($p < 0.006$).
4. In the other tests, although the recognition rate of 2DPCA was still better than that of PCA, the performance difference between PCA and 2DPCA was not statistically significant.

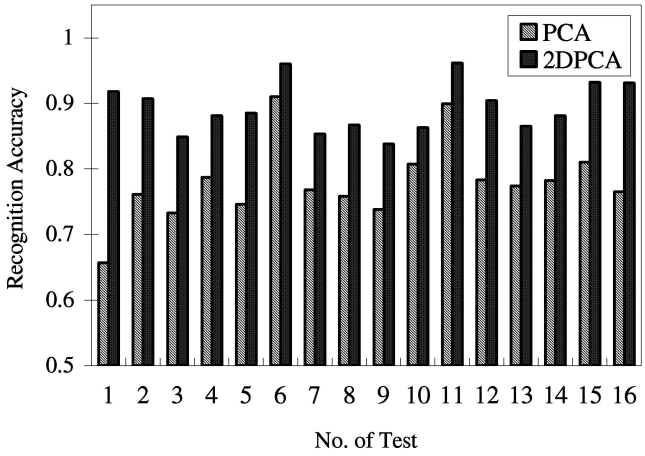

Fig. 9. Top recognition accuracy of 2DPCA and PCA corresponding to all of the 16 tests under varying illumination.

5 Conclusion and Future Work

In this paper, a new technique for image feature extraction and representation—two-dimensional principal component analysis (2DPCA)—was developed. 2DPCA has many advantages over conventional PCA (Eigenfaces). In the first place, since 2DPCA is based on the image matrix, it is simpler and more straightforward to use for image feature extraction. Second, 2DPCA is better than PCA in terms of recognition accuracy in all experiments. Although this trend seems to be consistent for different databases and conditions, in some experiments the differences in performance were not statistically significant. Third, 2DPCA is computationally

TABLE 5
Comparison of the Performance of 2DPCA, Eigenfaces, ICA, and Kernel Eigenfaces Using the Yale Database (Note that ICA Is Tested Using Euclidean Distance in [14])

Method	Recognition accuracy
Eigenfaces [14] *	71.52% (118/165)
ICA [14] *	71.52% (118/165)
Kernel Eigenfaces [14] *	72.73% (120/165)
2DPCA	84.24% (139/165)

Note that the asterisk indicates a statistically significant difference between the marked method and 2DPCA at a significance level of 0.05.

more efficient than PCA and it can improve the speed of image feature extraction significantly. However, it should be pointed out that 2DPCA-based image representation was not as efficient as PCA in terms of storage requirements, since 2DPCA requires more coefficients for image representation than PCA.

Why does 2DPCA outperform PCA in face recognition? In our opinion, the underlying reason is that 2DPCA is more suitable for small sample size problems (like face recognition) since its image covariance matrix is quite small. Image representation and recognition based on PCA (or 2DPCA) is statistically dependent on the evaluation of the covariance matrix (although for PCA the explicit construction of the covariance matrix can be avoided). The obvious advantage of 2DPCA over PCA is that the former evaluates the covariance matrix more accurately.

Finally, there are still some aspects of 2DPCA that deserve further study. When a small number of the principal components of PCA are used to represent an image, the mean square error (MSE) between the approximation and the original pattern is minimal. Does 2DPCA have a similar property? In addition, 2DPCA needs more coefficients for image representation than PCA. Although, as a feasible alternative to deal with this problem is to use PCA after 2DPCA for further dimensional reduction, it is still unclear how the dimension of 2DPCA could be reduced directly.

ACKNOWLEDGMENTS

This work is partially supported by Centre of Multimedia Signal Processing and the central fund from The Hong Kong Polytechnic University. And, it is partially supported by grants TIC2002-04495-C02 from the same Spanish Ministry of Science and Technology (MCyT) and AutenticUZ (UZ-2001-TEC-01) from the University of Zaragoza. It is also partially supported by the National Science Foundation of China under grant no. 60072034. Finally, the authors would like to thank the anonymous reviewers for their constructive advice.

REFERENCES

[1] L. Sirovich and M. Kirby, "Low-Dimensional Procedure for Characterization of Human Faces," *J. Optical Soc. Am.*, vol. 4, pp. 519-524, 1987.
[2] M. Kirby and L. Sirovich, "Application of the KL Procedure for the Characterization of Human Faces," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 103-108, Jan. 1990.
[3] M. Turk and A. Pentland, "Eigenfaces for Recognition," *J. Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.
[4] A. Pentland, "Looking at People: Sensing for Ubiquitous and Wearable Computing," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 107-119, Jan. 2000.
[5] M.A. Grudin, "On Internal Representations in Face Recognition Systems," *Pattern Recognition*, vol. 33, no. 7, pp. 1161-1177, 2000.
[6] G.W. Cottrell and M.K. Fleming, "Face Recognition Using Unsupervised Feature Extraction," *Proc. Int'l Neural Network Conf.*, pp. 322-325, 1990.
[7] D. Valentin, H. Abdi, A.J. O'Toole, and G.W. Cottrell, "Connectionist Models of Face Processing: a Survey," *Pattern Recognition*, vol. 27, no. 9, pp. 1209-1230, 1994.
[8] P.S. Penev and L. Sirovich, "The Global Dimensionality of Face Space," *Proc. Fourth IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pp. 264-270, 2000.
[9] L. Zhao and Y. Yang, "Theoretical Analysis of Illumination in PCA-Based Vision Systems," *Pattern Recognition*, vol. 32, no. 4, pp. 547-564, 1999.
[10] L. Wiskott, J.M. Fellous, N. Krüger, and C. von der Malsburg, "Face Recognition by Elastic Bunch Graph Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 775-779, July 1997.
[11] M.S. Bartlett, J.R. Movellan, and T.J. Sejnowski, "Face Recognition by Independent Component Analysis," *IEEE Trans. Neural Networks*, vol. 13, no. 6, pp. 1450-1464, 2002.
[12] B.A. Draper, K. Baek, M.S. Bartlett, J.R. Beveridge, "Recognizing Faces with PCA and ICA," *Computer Vision and Image Understanding: special issue on face recognition*, in press.
[13] P.C. Yuen and J.H. Lai, "Face Representation Using Independent Component Analysis," *Pattern Recognition*, vol. 35, no. 6, pp. 1247-1257, 2002.
[14] M.H. Yang, "Kernel Eigenfaces vs. Kernel Fisherfaces: Face Recognition Using Kernel Methods," *Proc. Fifth IEEE Int'l Conf. Automatic Face and Gesture Recognition (RGR'02)*, pp. 215-220, May 2002.
[15] K. Liu et al., "Algebraic Feature Extraction for Image Recognition Based on an Optimal Discriminant Criterion," *Pattern Recognition*, vol. 26, no. 6, pp. 903-911, 1993.
[16] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriengman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711-720, July 1997.
[17] A.M. Martinez and R. Benavente, "The AR Face Database," CVC Technical Report, no. 24, June 1998.
[18] A.M. Martinez and R. Benavente, "The AR Face Database," http://rvl1.ecn.purdue.edu/~aleix/aleix_face_DB.html, 2003.
[19] J. Yang, J.Y. Yang, "From Image Vector to Matrix: A Straightforward Image Projection Technique—IMPCA vs. PCA," *Pattern Recognition*, vol. 35, no. 9, pp. 1997-1999, 2002.
[20] W. Yambor, B. Draper, and R. Beveridge, "Analyzing PCA-Based Face Recognition Algorithms: Ei-Genvector Selection and Distance Measures," *Empirical Evaluation Methods in Computer Vision*, H. Christensen and J. Phillips, eds., Singapore: World Scientific Press, 2002.
[21] J.R. Beveridge, K. She, B. Draper, and G.H. Givens, "Parametric and Nonparametric Methods for the Statistical Evaluation of Human ID Algorithms," *Proc. Third Workshop Empirical Evaluation of Computer Vision Systems*, Dec. 2001.