BDPCA Plus LDA: A Novel Fast Feature Extraction Technique for Face Recognition

Wangmeng Zuo, David Zhang, *Senior Member, IEEE*, Jian Yang, and Kuanquan Wang

***Abstract*—Appearance-based methods, especially linear discriminant analysis (LDA), have been very successful in facial feature extraction, but the recognition performance of LDA is often degraded by the so-called "small sample size" (SSS) problem. One popular solution to the SSS problem is principal component analysis (PCA) + LDA (Fisherfaces), but the LDA in other low-dimensional subspaces may be more effective. In this correspondence, we proposed a novel fast feature extraction technique, bidirectional PCA (BDPCA) plus LDA (BDPCA + LDA), which performs an LDA in the BDPCA subspace. Two face databases, the ORL and the Facial Recognition Technology (FERET) databases, are used to evaluate BDPCA + LDA. Experimental results show that BDPCA + LDA needs less computational and memory requirements and has a higher recognition accuracy than PCA + LDA.**

***Index Terms*—Bidirectional principal component analysis (BDPCA), face recognition, feature extraction, linear discriminant analysis (LDA), principal component analysis (PCA).**

I. Introduction

Face recognition has been an important issue in computer vision and pattern recognition over the last several decades [4], [33]. While a human can recognize faces easily, automated face recognition remains a great challenge in computer-based automated recognition research. One difficulty in face recognition is how to handle the variations in expression, pose, and illumination when only a limited number of training samples are available.

Currently, face-recognition methods are of two types, the geometric-based approaches and the holistic-based approaches. Geometric-based approaches extract local features such as the locations and local statistics of the eyes, nose, and mouth and require correct feature detection and good measurement techniques. Holistic-based approaches extract a holistic representation of the whole face region and have a robust recognition performance under noise, blurring, and partial occlusion.

The two main holistic-based face-recognition approaches are the principal component analysis (PCA) and the linear discriminant analysis (LDA). PCA was first used in 1987 by Sirovich and Kirby to represent facial images [14], [23]. Subsequently, Turk and Pentland applied PCA to face recognition and presented the well-known eigenfaces method [26]. Since then, PCA has been widely studied and has become one of most successful facial-feature-extraction approaches. Recently, other PCA-based approaches, such as the two-dimensional (2-D) PCA (2DPCA), have also been proposed for face recognition [6], [11], [27], [30].

The second holistic-based face-recognition approach (the LDA) works by finding the set of the optimal projection vectors that map the original data into a low-dimensional feature space with the restriction that the ratio of the trace of the between-class scatter $\mathbf{S}_b$ to the trace of the within-class scatter $\mathbf{S}_w$ is maximized [8]. When applied to face recognition, the LDA seriously suffers from the so-called "small sample size" (SSS) problem caused by the limited number of high-dimensional training samples [5], [31].

A number of approaches have been proposed to address the SSS problem. One of the most successful approaches is subspace LDA, which uses a dimensionality reduction technique to map the original data to a low-dimensional subspace. Researchers have applied the PCA, latent semantic indexing (LSI) [7], and partial least squares (PLS) [9] as preprocessors for dimensionality reduction [1], [2], [24], [25], [32]. Among all the subspace LDA methods, over the past ten years, the PCA plus LDA approach (PCA + LDA) has received significant attention. In the Fisherfaces, PCA is first applied to eliminate the singularity of $\mathbf{S}_w$, and then LDA is performed in the PCA subspace [2]. However, the discarded null space of $\mathbf{S}_w$ may contain some important discriminant information and may cause the performance deterioration of the Fisherfaces. Rather than discarding the null space of $\mathbf{S}_w$, researchers proposed a class of direct LDA (D-LDA) method [5], [31], while Yang proposed a complete PCA + LDA method, which simultaneously considered the discriminant information both in the range space and the null space of $\mathbf{S}_w$ [28].

In this correspondence, we propose a fast subspace LDA technique, bidirectional PCA (BDPCA) plus LDA (BDPCA + LDA). The BDPCA, which assumes that the transform kernel of PCA is separable, is a natural extension of the classical PCA [13], [21] and a generalization of Yang's 2DPCA [27], [30]. The separation of the PCA kernel has at least three main advantages: faster training, faster feature extraction, and a lower memory requirement.

This correspondence is organized as follows. Section II briefly reviews previous work on the PCA technique applied to 2-D image transforms. Section III first proposes a separable 2-D transform technique (BDPCA) and then presents a fast face-recognition technique BDPCA + LDA. It also provides a detailed comparison of the PCA + LDA and the BDPCA + LDA frameworks. Section IV presents the results of experiments using the ORL database and the Facial Recognition Technology (FERET) database. Finally, Section V offers our conclusion.

II. Overview of PCA Techniques for a 2-D Image Transform

A. Two-Dimensional Image Transform

A 2-D image transform has two major applications in image processing: image feature extraction and image dimensionality reduction. In [13] and [21], a 2-D transform is defined as follows.

Definition 1: The 2-D transform of the $m \times n$ image matrix $\mathbf{X}(j,k)$ results in a transformed image matrix $\mathbf{X}'(u,v)$ as defined by

$$\mathbf{X}'(u,v) = \sum_{j=1}^{m}\sum_{k=1}^{n}\mathbf{X}(j,k)\mathbf{A}(j,k;u,v) \tag{1}$$

where $\mathbf{A}(i,j;u,v)$ denotes the transform kernel. The inverse transform is defined as

$$\widetilde{\mathbf{X}}(i,j) = \sum_{u=1}^{m}\sum_{v=1}^{n}\mathbf{X}'(u,v)\mathbf{B}(i,j;u,v) \tag{2}$$

where $\mathbf{B}(i,j;u,v)$ denotes the inverse transform kernel.

Manuscript received February 5, 2005; revised August 6, 2005. This work was supported in part by the Hong Kong Special Administrative Region Government under the University Grants Committee/Co-operative Research Centres fund, in part by Hong Kong Polytechnic University under the central fund, and in part by the National Natural Science Foundation of China under Contracts 60332010 and 90209020. This paper was recommended by Associate Editor X. Jiang.

W. Zuo and K. Wang are with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China (e-mail: cswmzuo@gmail.com; wangkq@hit.edu.cn).

D. Zhang and J. Yang are with the Biometrics Research Centre, Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: csdzhang@comp.polyu.edu.hk; csjyang@comp.polyu.edu.hk).

Digital Object Identifier 10.1109/TSMCB.2005.863377

Definition 2 [13], [21]: The transform is separable if its kernels can be rewritten as

$$\mathbf{A}(j,k;u,v) = \mathbf{A}_{\mathrm{col}}(j,u)\mathbf{A}_{\mathrm{row}}(k,v) \tag{3}$$

$$\mathbf{B}(j,k;u,v) = \mathbf{B}_{\mathrm{col}}(j,u)\mathbf{B}_{\mathrm{row}}(k,v). \tag{4}$$

The introduction of the term "separable" is very important in a 2-D transform. In a separable transform, the transformed matrix $\mathbf{X}'$ of the original image matrix $\mathbf{X}$ can be obtained by

$$\mathbf{X}' = \mathbf{A}_{\mathrm{col}}\mathbf{X}\mathbf{A}_{\mathrm{row}}^{\mathrm{T}} \tag{5}$$

and the inverse transformation can be given by

$$\widetilde{\mathbf{X}} = \mathbf{B}_{\mathrm{col}}\mathbf{X}'\mathbf{B}_{\mathrm{row}}^{\mathrm{T}} \tag{6}$$

where $\mathbf{A}_{\mathrm{col}}$ and $\mathbf{A}_{\mathrm{row}}$ are column and row transform kernels, respectively, and $\mathbf{B}_{\mathrm{col}}$ and $\mathbf{B}_{\mathrm{row}}$ are column and row inverse transform kernels, respectively. If the separable transform is unitary [21], the inverse transform kernels are then obtained by

$$\mathbf{B}_{\mathrm{col}} = \mathbf{A}_{\mathrm{col}}^{\mathrm{T}} \qquad \mathbf{B}_{\mathrm{row}} = \mathbf{A}_{\mathrm{row}}^{\mathrm{T}}. \tag{7}$$

B. Holistic PCA: Nonseparable Image Model-Based Technique

When handling a nonseparable 2-D transform, it is better to map the image matrix $\mathbf{X}(j,k)$ into its vector representation $\mathbf{x}$ in advance. Then, the 2-D transform of (1) can be rewritten as

$$\mathbf{x}' = \mathbf{A}\mathbf{x}. \tag{8}$$

A holistic PCA transform, also known as the Karhunen–Loeve transform (KLT), is an important nonseparable 2-D image transform technique. In a holistic PCA, an image matrix $\mathbf{X}$ must be transformed into a one-dimensional (1-D) vector $\mathbf{x}$ in advance. Then, given a set of N training images $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, the total scatter matrix $\mathbf{S}_{\mathrm{t}}$ of PCA is defined by

$$\mathbf{S}_{\mathrm{t}} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^{\mathrm{T}} \tag{9}$$

where $\overline{\mathbf{x}}$ denotes the mean vector of all training images.

We then choose eigenvectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_i, \ldots, \mathbf{v}_{d_{\mathrm{PCA}}}\}$ corresponding to the first d_{PCA} largest eigenvalues $\{\lambda_1, \ldots, \lambda_i, \ldots, \lambda_{d_{\mathrm{PCA}}}\}$ of $\mathbf{S}_{\mathrm{t}}$ as projection axes. After the projection of sample $\mathbf{x}$ onto the eigenvector $\mathbf{v}_i$

$$y_i = \mathbf{v}_i^{\mathrm{T}}(\mathbf{x} - \overline{\mathbf{x}}), \qquad i = 1, \ldots, d_{\mathrm{PCA}} \tag{10}$$

we can form the PCA-transformed vector $\mathbf{y} = [y_1, y_2, \ldots, y_{d_{\mathrm{PCA}}}]^{\mathrm{T}}$ of sample $\mathbf{x}$.

C. Two-dimensional KLT (2D-KLT): Separable Image Model-Based Technique

The 2D-KLT is a separable PCA technique. If the PCA kernel is separable, we can rewrite the 2-D transform of (1) as

$$\mathbf{X}' = \mathbf{A}_{\mathrm{col}}^{\mathrm{T}}\mathbf{X}\mathbf{A}_{\mathrm{row}} \tag{11}$$

where $\mathbf{A}_{\mathrm{row}}$ and $\mathbf{A}_{\mathrm{col}}$ are the row and column kernels that satisfy

$$\mathbf{S}_{\mathrm{t}}^{\mathrm{col}}\mathbf{A}_{\mathrm{col}} = \mathbf{\Lambda}_{\mathrm{col}}\mathbf{A}_{\mathrm{col}} \tag{12}$$

$$\mathbf{S}_{\mathrm{t}}^{\mathrm{row}}\mathbf{A}_{\mathrm{row}} = \mathbf{\Lambda}_{\mathrm{row}}\mathbf{A}_{\mathrm{row}} \tag{13}$$

where $\mathbf{S}_{\mathrm{t}}^{\mathrm{col}}$ and $\mathbf{S}_{\mathrm{t}}^{\mathrm{row}}$ are the column and row total covariance matrices, respectively, and $\mathbf{\Lambda}_{\mathrm{col}}$ and $\mathbf{\Lambda}_{\mathrm{row}}$ are two diagonal matrices.

Based on the assumption that the column total scatter matrix $\mathbf{S}_{\mathrm{t}}^{\mathrm{col}}$ is defined by a first-order Markov process with a correlation factor r, Habibi and Wintz [12] and Ray and Driver [22] gave the column eigenvalues λ_k^{col} and eigenvectors $\boldsymbol{\psi}_k^{\mathrm{col}} = [\boldsymbol{\psi}_k^{\mathrm{col}}(1), \ldots, \boldsymbol{\psi}_k^{\mathrm{col}}(i), \ldots, \boldsymbol{\psi}_k^{\mathrm{col}}(m)]^{\mathrm{T}}$ as

$$\lambda_k^{\mathrm{col}} = \frac{1 - r^2}{1 - 2r\cos\omega_k^{\mathrm{col}} + r^2} \tag{14}$$

$$\boldsymbol{\psi}_k^{\mathrm{col}}(i) = \left(\frac{2}{m + \lambda_k^{\mathrm{col}}}\right)^{\frac{1}{2}} \sin\left[\omega_k^{\mathrm{col}}\left(i - \frac{(m+1)\pi}{2} + \frac{k\pi}{2}\right)\right]$$
$$i = 1, \ldots, m \tag{15}$$

where ω_k^{col} are the real positive roots of the transcendental equation for $m =$ even

$$\tan(m\omega) = \frac{(1 - r^2)\sin\omega}{\cos\omega - 2r + r^2\cos\omega}. \tag{16}$$

Similarly, we can calculate the row eigenvalues λ_k^{row} and eigenvectors $\boldsymbol{\psi}_k^{\mathrm{row}}$.

Then, we compare the computational complexity of the holistic PCA and the 2D-KLT. It is reasonable to use the number of multiplications as a measurement of computational complexity of PCA and 2D-KLT transforms. The holistic PCA transform requires $(mn)^2$ multiplications whereas the 2D-KLT transform requires $m^2n + n^2m$ multiplications.

Yang recently proposed another 2-D transform, 2DPCA [27], [30]. Note that 2DPCA is different from 2D-KLT in two ways. First, 2D-KLT uses $\mathbf{X}' = \mathbf{A}_{\mathrm{col}}^{\mathrm{T}}\mathbf{X}\mathbf{A}_{\mathrm{row}}$ to transform an image $\mathbf{X}$ to $\mathbf{X}'$, whereas 2DPCA uses $\mathbf{X}' = \mathbf{X}\mathbf{A}_{\mathrm{row}}$. Second, 2DPCA and 2D-KLT define the row total scatter matrix $\mathbf{S}_{\mathrm{t}}^{\mathrm{row}}$ differently.

III. BDPCA Plus LDA: A New Strategy for Facial Feature Extraction

A. BDPCA: A Face-Image-Specific 2D-KLT Technique

In this section, we propose a face-image-specific transform, the BDPCA. Given a training set $\{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$, N is the number of the training images and the size of each image matrix is $m \times n$. By representing the ith image matrix $\mathbf{X}_i$ as an m set of $1 \times n$ row vectors

$$\mathbf{X}_i = \begin{bmatrix} \mathbf{x}_i^1 \\ \mathbf{x}_i^2 \\ \vdots \\ \mathbf{x}_i^m \end{bmatrix} \tag{17}$$

we adopt Yang's approach [27], [30] to define the row total scatter matrix

$$\mathbf{S}_{\mathrm{t}}^{\mathrm{row}} = \frac{1}{Nm}\sum_{i=1}^{N}\sum_{j=1}^{m}\left(\mathbf{x}_i^j - \overline{\mathbf{x}}^j\right)^{\mathrm{T}}\left(\mathbf{x}_i^j - \overline{\mathbf{x}}^j\right)$$
$$= \frac{1}{Nm}\sum_{i=1}^{N}(\mathbf{X}_i - \overline{\mathbf{X}})^{\mathrm{T}}(\mathbf{X}_i - \overline{\mathbf{X}}) \tag{18}$$

where $\mathbf{x}_i^j$ and $\overline{\mathbf{x}}_i^j$ denote the jth row of sample $\mathbf{X}_i$ and mean matrix $\overline{\mathbf{X}}$, respectively. We choose the row eigenvectors corresponding to the

TABLE I
COMPARISONS OF COMPUTATIONAL AND MEMORY REQUIREMENTS OF THE BDPCA + LDA AND THE PCA + LDA

Method	Memory Requirements		Computation Requirements	
	Projector	Feature prototypes	Training	Testing
PCA+LDA	$(m\times n)\times d_{\mathrm{LDA}}$ **Large**	$N\times d_{\mathrm{LDA}}$ **Same**	a) Calculating the projector: $\mathrm{O}(N_{\mathrm{p}}^3+d_{\mathrm{PCA}}^3)$ **Large** b) Projection: $N\times(m\times n)\times d_{\mathrm{LDA}}$ **Large**	c) Projection: $(m\times n)\times d_{\mathrm{LDA}}$ **Large** d) Distance calculation: $N\times d_{\mathrm{LDA}}$ **Same**
BDPCA +LDA	$m\times k_{\mathrm{row}}+n\times k_{\mathrm{col}}+k_{\mathrm{col}}\times k_{\mathrm{row}}\times d_{\mathrm{LDA}}$ **Small**	$N\times d_{\mathrm{LDA}}$ **Same**	a) Calculating the projector: $\mathrm{O}(m^3+n^3+d_{\mathrm{BDPCA}}^3)$ **Small** b) Projection: $N\times[m\times n\times\min(k_{\mathrm{row}}, k_{\mathrm{col}})+k_{\mathrm{col}}\times k_{\mathrm{row}}\times\max(m+d_{\mathrm{LDA}}, n+d_{\mathrm{LDA}})]$ **Small**	c) Projection: $m\times n\times\min(k_{\mathrm{row}}, k_{\mathrm{col}})+k_{\mathrm{col}}\times k_{\mathrm{row}}\times[\max(m,n)+d_{\mathrm{LDA}}]$ **Small** d) Distance calculation: $N\times d_{\mathrm{LDA}}$ **Same**

first k_{row} largest eigenvalues of $\mathbf{S}_{\mathrm{t}}^{\mathrm{row}}$ to construct the row projection matrix $\mathbf{W}_r$

$$\mathbf{W}_r = \left[\mathbf{w}_1^{\mathrm{row}}, \mathbf{w}_2^{\mathrm{row}}, \ldots, \mathbf{w}_{k_{\mathrm{row}}}^{\mathrm{row}}\right] \tag{19}$$

where $\mathbf{w}_i^{\mathrm{row}}$ denotes the row eigenvector corresponding to the ith largest eigenvalues of $\mathbf{S}_{\mathrm{t}}^{\mathrm{row}}$.

Similarly, by treating an image matrix $\mathbf{X}_i$ as an n set of $m \times 1$ column vectors

$$\mathbf{X}_i = \begin{bmatrix}\mathbf{x}_i^1 & \mathbf{x}_i^2 & \cdots & \mathbf{x}_i^n\end{bmatrix} \tag{20}$$

we define the column total scatter matrix

$$\mathbf{S}_{\mathrm{t}}^{\mathrm{col}} = \frac{1}{Nn}\sum_{i=1}^{N}(\mathbf{X}_i - \overline{\mathbf{X}})(\mathbf{X}_i - \overline{\mathbf{X}})^{\mathrm{T}}. \tag{21}$$

We then choose the column eigenvectors corresponding to the first k_{col} largest eigenvalues of $\mathbf{S}_{\mathrm{t}}^{\mathrm{col}}$ to construct the column projection matrix $\mathbf{W}_c$

$$\mathbf{W}_c = \left[\mathbf{w}_1^{\mathrm{col}}, \mathbf{w}_2^{\mathrm{col}}, \ldots, \mathbf{w}_{k_{\mathrm{col}}}^{\mathrm{col}}\right] \tag{22}$$

where $\mathbf{w}_i^{\mathrm{col}}$ is the column eigenvector corresponding to the ith largest eigenvalues of $\mathbf{S}_{\mathrm{t}}^{\mathrm{col}}$.

Finally we use the transformation

$$\mathbf{Y} = \mathbf{W}_c^{\mathrm{T}}\mathbf{X}\mathbf{W}_r \tag{23}$$

to extract the feature matrix $\mathbf{Y}$ of image matrix $\mathbf{X}$.

B. BDPCA Plus LDA Technique

In this section, we propose a BDPCA plus LDA technique for fast facial feature extraction. The BDPCA + LDA is an LDA approach that is applied on a low-dimensional BDPCA subspace. Since less time is required to map an image matrix to BDPCA subspace, the BDPCA + LDA is, at least, computationally faster than the PCA + LDA.

The BDPCA + LDA first uses a BDPCA to obtain feature matrix $\mathbf{Y}$ as

$$\mathbf{Y} = \mathbf{W}_c^{\mathrm{T}}\mathbf{X}\mathbf{W}_r \tag{24}$$

where $\mathbf{W}_c$ and $\mathbf{W}_r$ are the column and row projectors, respectively; $\mathbf{X}$ is an image matrix; and $\mathbf{Y}$ is its BDPCA feature matrix. The feature matrix $\mathbf{Y}$ is then transformed into feature vector $\mathbf{y}$ by concatenating the columns of $\mathbf{Y}$. The LDA projector $\mathbf{W}_{\mathrm{LDA}} = [\varphi_1, \varphi_2, \ldots, \varphi_m]$ is calculated by maximizing Fisher's criterion

Fig. 1. Five images of an individual from the ORL face database.

$$J(\varphi) = \frac{\varphi^{\mathrm{T}}\mathbf{S}_b\varphi}{\varphi^{\mathrm{T}}\mathbf{S}_w\varphi} \tag{25}$$

where φ_i is the generalized eigenvector of $\mathbf{S}_b$ and $\mathbf{S}_w$ corresponding to the ith largest eigenvalue λ_i

$$\mathbf{S}_b\varphi_i = \lambda_i\mathbf{S}_w\varphi_i. \tag{26}$$

$\mathbf{S}_b$ is the between-class scatter matrix of $\mathbf{y}$

$$\mathbf{S}_b = \frac{1}{N}\sum_{i=1}^{C}N_i(\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^{\mathrm{T}} \tag{27}$$

and $\mathbf{S}_w$ is the within-class scatter matrix of $\mathbf{y}$

$$\mathbf{S}_w = \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{N_i}(\mathbf{y}_{i,j} - \boldsymbol{\mu}_i)(\mathbf{y}_{i,j} - \boldsymbol{\mu}_i)^{\mathrm{T}} \tag{28}$$

where N_i, $\mathbf{y}_{i,j}$, and $\boldsymbol{\mu}_i$ are the number of feature vectors, the jth feature vector, and the mean vector of class i, respectively. C is the number of classes, and $\boldsymbol{\mu}$ is the mean vector of all the feature vectors.

In summary, the main steps in BDPCA + LDA feature extraction are to first transform an image matrix $\mathbf{X}$ into BDPCA feature subspace $\mathbf{Y}$ by (24), to map $\mathbf{Y}$ into its 1D representation $\mathbf{y}$, and then to obtain the final feature vector $\mathbf{z}$ by

$$\mathbf{z} = \mathbf{W}_{\mathrm{LDA}}^{\mathrm{T}}\mathbf{y}. \tag{29}$$

C. Advantages Over the Existing PCA Plus LDA Framework

In this section, we compare the BDPCA + LDA and the PCA + LDA face-recognition frameworks in terms of their computational and memory requirements. It is worth noting that the computational requirements are considered in two phases, training and testing.

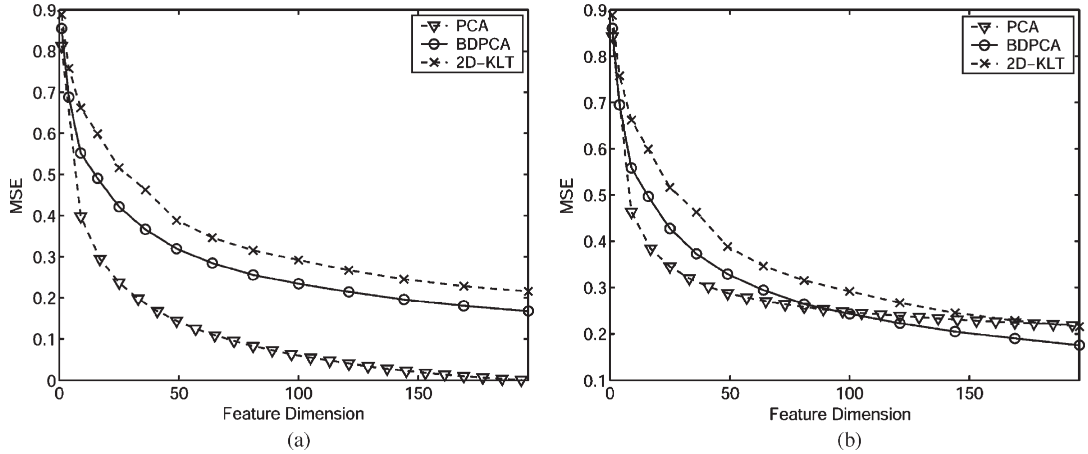

Fig. 2. MSE curves of the PCA, the 2D-KLT, and the BDPCA (a) on the training set and (b) on the testing set.

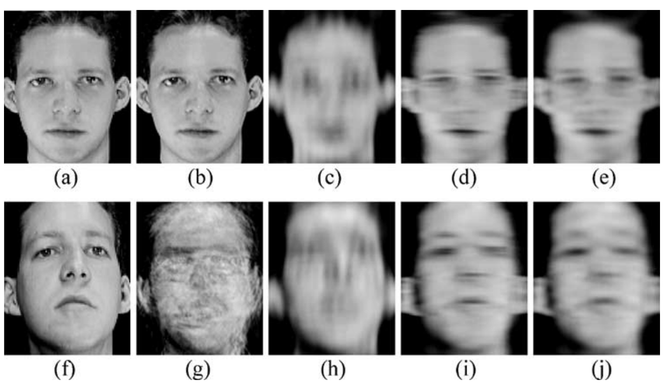

Fig. 3. Comparisons of the reconstruction capability of the four methods. (a) and (f) Original images. Reconstructed images by (b) and (g) the PCA, (c) and (h) the 2D-KLT, (d) and (i) the 2DPCA, and (e) and (j) the BDPCA.

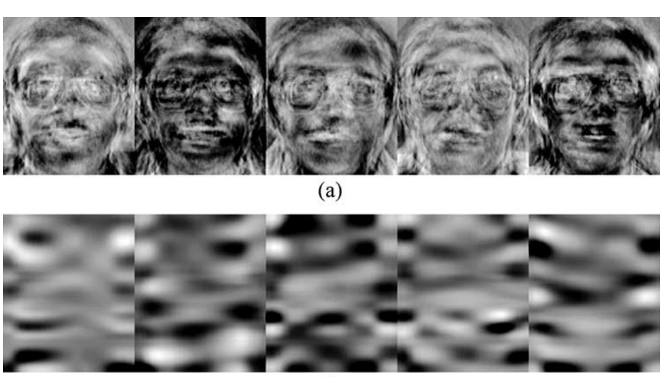

Fig. 4. Example of the discriminant vectors of (a) the Fisherfaces and (b) the BDPCA + LDA.

TABLE II
COMPARISONS OF THE ARR OBTAINED USING THE BDPCA WITH DIFFERENT PARAMETERS

k_{row} \ k_{col}	1	6	12	18	24	30	112
1	0.138	0.851	0.905	0.924	0.920	0.920	0.919
4	0.616	0.942	0.949	**0.951**	0.949	0.950	0.950
8	0.719	0.937	0.941	0.941	0.943	0.941	0.940
12	0.734	0.941	0.943	0.942	0.942	0.942	0.941
16	0.742	0.936	0.944	0.941	0.942	0.941	0.940
20	0.741	0.934	0.942	0.944	0.943	0.942	0.940
92	0.741	0.936	0.944	0.945	0.944	0.942	0.941

TABLE III
COMPARISONS OF THE ARR OBTAINED USING THE BDPCA + LDA WITH DIFFERENT PARAMETERS

k_{row} \ k_{col}	4	8	10	12	15	20	25
2	0.937	0.957	0.957	0.954	0.958	0.944	0.940
3	0.932	0.957	0.958	0.967	0.957	0.948	0.940
4	0.947	0.958	0.970	**0.971**	0.968	0.959	0.939
6	0.954	0.955	0.968	0.960	0.950	0.933	0.896
8	0.958	0.955	0.965	0.956	0.940	0.906	–
10	0.944	0.954	0.959	0.934	–	–	–
12	0.940	0.942	0.938	0.864	–	–	–

TABLE IV
TOTAL CPU TIME (SECONDS) FOR TRAINING AND TESTING ON THE ORL DATABASE

Method	Time for Training (s)	Time for Testing (s)
PCA+LDA	46.0	5.2
BDPCA+LDA	17.5	3.9

We first compare the computational requirement using the number of multiplications as a measurement of computational complexity. The training phase involves two computational tasks: 1) calculation of the projector and 2) projection of images into feature prototypes. To calculate the projector, the PCA + LDA method must solve an $N \times N$ eigenvalue problem and then a $d_{\mathrm{PCA}} \times d_{\mathrm{PCA}}$ generalized eigenvalue problem, where N is the size if the training set and d_{PCA} is the dimension of the PCA subspace. In contrast, BDPCA + LDA must solve an $m \times m$ problem, an $n \times n$ eigenvalue problem, and a $d_{\mathrm{BDPCA}} \times d_{\mathrm{BDPCA}}$ generalized eigenvalue problem, where d_{BDPCA} is the dimension of BDPCA subspace. Since the complexity of an $M \times M$ eigenvalue problem is $\mathrm{O}(M^3)$ [10], the complexity of the PCA + LDA projector-calculation operation is $\mathrm{O}(N^3 + d_{\mathrm{PCA}}^3)$ whereas that of BDPCA + LDA is $\mathrm{O}(m^3 + n^3 + d_{\mathrm{BDPCA}}^3)$. Assuming that m, n, d_{PCA}, and d_{BDPCA} are smaller than the number of training samples N, BDPCA + LDA requires less computation than PCA + LDA to calculate the projector. In Section IV, this assumption is satisfied in all the experiments.

TABLE V
COMPARISONS OF COMPUTATIONAL AND MEMORY REQUIREMENTS OF THE BDPCA + LDA AND THE PCA + LDA ON THE ORL DATABASE

Method	Memory Requirements			Computation Requirements	
	Projector	Feature prototypes	Total	Training	Testing
PCA+LDA	(112×92)×39 =*401856*	200×39 =*7800*	*409656*	a) Calculating the projector: O(200^3+160^3) ≈*12096000* b) Projection: 200×(112×92)×39 =*80371200* Total=*92467200*	c) Projection: (112×92)×39 =*401856* d) Distance calculation: 200×39 =*7800* Total=*409656*
BDPCA +LDA	112×12+92×4 +(12×4)×39 =*3584*	200×39 =*7800*	*11384*	a) Calculating the projector: O(112^3+92^3+$(12×4)^3$) ≈*2294208* b) Projection: 200×[112×92×4 +12×4 ×(112+39)]=*9692800* Total=*11987008*	c) Projection: 112×92×4+4×12 ×(112+39) =*48464* d) Distance calculation: 200×39 =*7800* Total=*56264*

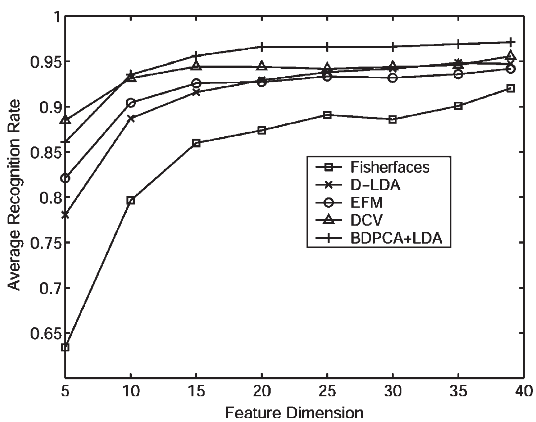

Fig. 5. Comparisons of the recognition rates obtained using different methods on the ORL database.

TABLE VI
RECOGNITION PERFORMANCE OF FIVE FACE RECOGNITION METHODS ON THE ORL DATABASE

Methods	Fisherfaces	D-LDA	EFM	DCV	BDPCA+LDA
Parameters	[d_{PCA}, d]	[d_{b}, d]	[d_{PCA}, d_{w}, d]	[d]	[k_{col}, k_{row}, d]
Values	[160, 39]	[39, 39]	[100, 100, 39]	[39]	[12, 4, 39]
ARR (%)	92.05%	94.20%	94.72%	95.55%	97.10%

TABLE VII
OTHER RESULTS RECENTLY REPORTED ON THE ORL DATABASE

Methods	Recognition Rate (%)	Year
Complete PCA+LDA [28]	97.0%	2003
DF-LDA [17]	95.8%	2003
NKFDA [15]	95.1%	2004
ELDA [34]	95.85%	2004
BDPCA+LDA	97.10%	

To project images into feature prototypes, we assume that the feature dimension of BDPCA + LDA and PCA + LDA is the same, d_{LDA}. For PCA + LDA, the number of multiplications is thus $N_{\mathrm{p}} \times (m \times n) \times d_{\mathrm{LDA}}$. For BDPCA + LDA, the number of multiplications is less than $N_{\mathrm{p}} \times [m \times n \times \min(k_{\mathrm{row}}, k_{\mathrm{col}}) + (k_{\mathrm{col}} \times k_{\mathrm{row}}) \times \max(m + d_{\mathrm{LDA}}, n + d_{\mathrm{LDA}})]$, where N_{p} is the number of prototypes. In this correspondence, we use all the prototypes for training; thus, $N_{\mathrm{p}} = N$. Assuming that $\min(k_{\mathrm{row}}, k_{\mathrm{col}})$ is much less than d_{LDA}, in the projection process, BDPCA + LDA also requires less computation than PCA + LDA. In Section IV, we will show that this assumption is satisfied in all our experiments.

In the test phase, there are two computational tasks: 1) the projection of images into the feature vector and 2) the calculation of the distance between the feature vector and feature prototypes. In the following, we compare the computational requirement of the BDPCA + LDA and the PCA + LDA in carrying out these two tasks. When projecting images into feature vectors, the BDPCA + LDA requires less computation than the PCA + LDA. Because the feature dimension of BDPCA + LDA and PCA + LDA is the same, the computational complexity of BDPCA + LDA and PCA + LDA are equal in the similarity measure process. Taking these two tasks into account, BDPCA + LDA is also less computationally expensive than PCA + LDA in the testing phase.

The memory requirements of the PCA + LDA and the BDPCA + LDA frameworks mainly depend on the size of the projector and the total size of the feature prototypes. The size of the projector of PCA + LDA is $d_{\mathrm{LDA}} \times m \times n$. This is because the PCA + LDA projector contains d_{LDA} Fisherfaces, each of which is of the same size as the original image. The BDPCA + LDA projector is in three parts, $\mathbf{W}_c$, $\mathbf{W}_r$, and $\mathbf{W}_{\mathrm{LDA}}$. The total size of the BDPCA + LDA projector is $(k_{\mathrm{col}} \times m) + (k_{\mathrm{row}} \times n) + (d_{\mathrm{LDA}} \times k_{\mathrm{col}} \times k_{\mathrm{row}})$, which is generally much smaller than that of the PCA + LDA. Finally, because these two methods have the same feature dimensions, BDPCA + LDA and PCA + LDA have equivalent feature prototype memory requirements.

We have compared the computational and the memory requirements of the BDPCA + LDA and the PCA + LDA frameworks, as listed in Table I. Generally, the BDPCA + LDA framework is superior to the PCA + LDA in both the computational and the memory requirements.

IV. EXPERIMENTS AND ANALYSIS

To test the efficacy of BDPCA + LDA, we make use of two face databases, the ORL face database [18] and the FERET database [19], [20]. We also compare the BDPCA + LDA with other LDA-based methods, including the Fisherfaces [2], enhanced Fisher discriminant model (EFM) [16], discriminant common vectors (DCV) [3], and D-LDA [31].

The experimental setup is as follows. Since our aim is to evaluate the efficacy of feature extraction methods, we use a simple classifier, the nearest neighbor classifier. To reduce the variation of recognition results, we adopt the mean of ten runs as the average recognition rate (ARR). All the experiments are carried out on an AMD 2500+ computer with a 512-Mb RAM and tested on the matrix laboratory (Matlab) platform (Version 6.5).

Fig. 6. Seven images of an individual from the FERET face database.

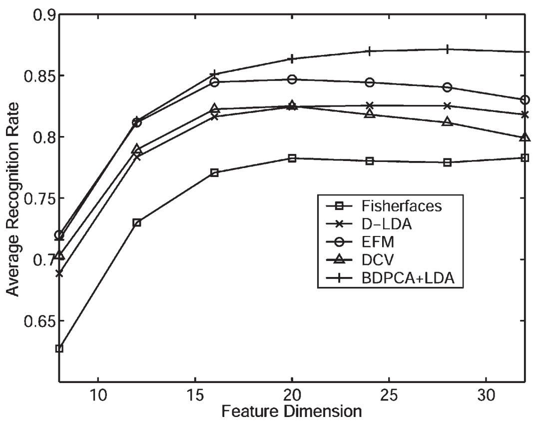

Fig. 7. Comparisons of the recognition rates obtained using different methods on the FERET database.

A. Experiments on the ORL Database

We use the ORL face database to test the performance of BDPCA + LDA in dealing with small variations in expressions, scale, and pose. The ORL database contains 400 facial images with ten images of each person. Fig. 1 shows the five images of an individual. The images have been collected at different times, under various light conditions, and with various facial expressions, facial details (glasses/no glasses), and variations in scale and tilt [18]. In order to evaluate the BDPCA representation and reconstruction, we compare the mean-square errors (MSE) of the 2D-KLT and the BDPCA and compare the reconstruction performances of the PCA, the 2D-KLT, the 2DPCA, and the BDPCA. Then, we present an intuitive illustration of the discriminant vectors obtained using the BDPCA + LDA. Finally, we carry out a comparison analysis of the BDPCA + LDA and other LDA-based methods.

To evaluate BDPCA representation and reconstruction, we construct a training set by choosing the first five images of each person. The remaining five images of each person are used for testing. Using this training set and this testing set, we test the MSE of BDPCA and intuitively evaluate the BDPCA reconstruction performance.

We compare the image representation performance of the PCA, the 2D-KLT, and the BDPCA according to their MSE curves on the training set and the testing set. Fig. 2(a) shows the plots of MSE curves on the training set. The MSE of the BDPCA is lower than that of the 2D-KLT, but the MSE of the PCA is much lower than those of the 2D-KLT and the BDPCA. This shows that the face-image-specific BDPCA represents facial images more effectively than the content-independent 2D-KLT, and the PCA is the best for representing training images. Fig. 2(b) shows the plots of MSE curves on the testing set. The MSE of BDPCA is also lower than that of the 2D-KLT, and the MSE of the PCA will be lower than that of the BDPCA when the feature dimension is higher than 90. This shows that the BDPCA is better than the 2D-KLT in facial image representation and is more robust than the PCA in representing the testing images.

We now intuitively compare the PCA, the 2D-KLT, the 2DPCA, and the BDPCA image reconstructions. Fig. 3 shows two original facial images and their reconstructions. Fig. 3(a) is a training image. We can see in Fig. 3(b)–(e) that the quality of the PCA reconstruction is perfect, the quality of the 2DPCA and BDPCA reconstruction is satisfactory, and the quality of the 2D-KLT is less satisfactory.

TABLE VIII
RECOGNITION PERFORMANCE OF FIVE FACE RECOGNITION METHODS ON THE FERET DATABASE

Methods	Fisherfaces	D-LDA	EFM	DCV	BDPCA+LDA
Parameters	[d_{PCA}, d]	[d_{b}, d]	[d_{PCA}, d_{w}, d]	[d]	[k_{col}, k_{row}, d]
Values	[200, 20]	[60, 24]	[100, 100, 24]	[20]	[15, 5, 28]
ARR (%)	78.26%	84.69%	84.69%	82.51%	87.14%

TABLE IX
TOTAL CPU TIME (SECONDS) FOR TRAINING AND TESTING ON THE FERET DATABASE

Method	Time for Training (s)	Time for Testing (s)
PCA+LDA	254.2	36.2
BDPCA+LDA	57.5	26.3

Fig. 3(f) is a facial image from the testing set. Fig. 3(g)–(j) shows the reconstructed images. We can see that the quality of the classical PCA has greatly deteriorated, but the 2DPCA and the BDPCA still perform well and the 2D-KLT is once again less satisfactory. It is worth pointing out that the feature dimension of 2DPCA is 896 (112×8), much higher than that of the PCA (180), the BDPCA $(30 \times 8 = 240)$, and the 2D-KLT (240).

Fig. 4 presents an intuitive illustration of the Fisherfaces and BDPCA + LDA's discriminant vectors. Fig. 4(a) shows the first five discriminant vectors obtained using the Fisherfaces, and Fig. 4(b) depicts the first five discriminant vectors obtained using BDPCA + LDA. It can be observed that the appearance of Fisherfaces' discriminant vectors is distinctly different from that of BDPCA + LDA. This establishes the novelty of the proposed LDA-based facial-feature-extraction technique.

In the following experiments on the ORL database, five images of each person are randomly chosen for training while the remaining five images are used for testing, producing a training set of 200 images and a testing set of 200 images. In this way, we run the face-recognition method (for example, BDPCA + LDA) ten times and calculate the ARR to reduce the variation of recognition rate.

We study the effect of parameter values on the recognition performance of the BDPCA and the BDPCA + LDA. Both the BDPCA and the BDPCA + LDA introduce two parameters, the number of column eigenvectors k_{col} and the number of row eigenvectors k_{row}. Table II depicts the effect of k_{col} and k_{row} on the ARR obtained using the BDPCA. As Table II shows, the maximum ARR is obtained when $k_{\text{row}} = 4$. When the number of column eigenvectors $k_{\text{col}} > 12$, k_{col} has little effect on the ARR of the BDPCA. Table III shows the effect of k_{col} and k_{row} on the ARR obtained using the BDPCA + LDA. As Table III shows, the maximum ARR (97.1%) is obtained when $k_{\text{row}} = 4$ and $k_{\text{col}} = 12$.

Table IV shows the total CPU time of the PCA + LDA (Fisherfaces) and the BDPCA + LDA in the training phase and in the testing phase. The BDPCA + LDA is much faster than the Fisherfaces both for training and for testing.

We compare the computational and the memory requirements of the BDPCA + LDA and the PCA + LDA (Fisherfaces). In Section III-C, based on a number of assumptions, we assert that the BDPCA + LDA is superior to the PCA + LDA in the computational and the memory requirements. We check the correctness of these assumptions. The size

TABLE X
COMPARISONS OF COMPUTATIONAL AND MEMORY REQUIREMENTS OF THE BDPCA + LDA AND THE PCA + LDA ON THE FERET DATABASE

Method	Memory Requirements			Computation Requirements	
	Projector	Feature prototypes	Total	Training	Testing
PCA+LDA	(112×92)×39 =*401856*	200×39 =*7800*	*409656*	a) Calculating the projector: $O(200^3+160^3)$ ≈*12096000* b) Projection: 200×(112×92)×39 =*80371200* Total=*92467200*	c) Projection: (112×92)×39=*401856* d) Distance calculation: 200×39 =*7800* Total=*409656*
BDPCA +LDA	112×12+92×4 +(12×4)×39 =*3584*	200×39 =*7800*	*11384*	a) Calculating the projector: $O(112^3+92^3+(12×4)^3)$ ≈*2294208* b) Projection: 200×[112×92×4+12×4 ×(112+39)]=*9692800* Total=*11987008*	c) Projection: 112×92×4+4×12 ×(112+39) =*48464* d) Distance calculation: 200×39 =*7800* Total=*56264*

of the training set is 200, higher than the size of the row vector (92) or the column vector (112). The feature dimension of the Fisherfaces is 39, much higher than the k_{row} (4). Thus, all these assumptions are satisfied. Table V shows the computational and the memory requirements of the BDPCA + LDA and the Fisherfaces. The BDPCA + LDA has a lower total memory requirement than the Fisherfaces and also requires less computation for training and testing.

We compare the recognition performance of the BDPCA + LDA with other feature extraction methods. Fig. 5 shows the ARR obtained using the BDPCA + LDA, the Fisherfaces, the EFM, the DCV, and the D-LDA. Table VI lists the optimal parameter values and the maximum ARR of these five methods. The maximum ARR obtained using BDPCA + LDA is 97.10%, higher than the ARRs obtained using the other four methods.

We also compare the recognition rate of the BDPCA + LDA with some recently reported results. Table VII lists the reported recognition rates obtained using other LDA-based methods on the ORL database, where the training set consisted of five images of each person. Note that some results were evaluated based on the performance of just one run [28] and that some results were evaluated based on the ARR of five runs or ten runs [15], [17]. Table VII shows that the BDPCA + LDA is very effective and competitive in facial feature extraction.

B. Experiments on the FERET Database

The FERET face database is sponsored by the U.S. Department of Defense and is one of the standard databases used in testing and in evaluating face-recognition algorithms [19], [20]. For our experiments, we chose a subset of the FERET database. This subset includes 1400 images of 200 individuals (each individual contributing seven images). The seven images of each individual consist of three front images, with varied facial expressions and illuminations, and four profile images ranging from $\pm 15^\circ$ to $\pm 25^\circ$ pose [29]. The facial portion of each original image was cropped to a size of 80 × 80 and preprocessed using a histogram equalization. Fig. 6 presents seven cropped images of one person.

In our experiments, three images of each person are randomly chosen for training, while the remaining four images are used for testing. Thus, we obtain a training set of 600 images and a testing set of 800 images. In this way, we run the face-recognition method ten times and calculate the ARR.

We compare the recognition rates obtained using the BDPCA + LDA, the Fisherfaces, the EFM, the DCV, and the D-LDA as shown in Fig. 7. We also list the optimal parameter values of each method and its maximum ARR in Table VIII. The maximum ARR of the BDPCA + LDA is 87.14%, higher than the ARRs of the other four methods.

Table IX shows the total CPU time of the PCA + LDA (EFM) and the BDPCA + LDA in the training phase and in the testing phase. The BDPCA + LDA is much faster than the EFM in both the training and the testing phases.

We compare the computational and the memory requirements of the BDPCA + LDA and the PCA + LDA (EFM). In Section III-C, based on a number of assumptions, we assert that the BDPCA + LDA is superior to the PCA + LDA in the computational and the memory requirements. We then check the correctness of these assumptions. The size of the training set is 600, much higher than the size of the row vector (80) or the column vector (80). The feature dimension of the EFM is 24, much higher than k_{row} (5). Thus, all these assumptions are satisfied. Table X shows the computational and the memory requirements of the BDPCA + LDA and the EFM. The BDPCA + LDA needs less computational and memory requirements than the EFM.

Comparing Table IX with Table IV, much more training time is saved by the BDPCA + LDA framework for the FERET database. This is because the training complexity of the BDPCA + LDA is $\mathrm{O}(N)$, whereas that of the PCA + LDA is $\mathrm{O}(N^3)$, where N is the size of training set. This property implies that, when the size of the training set is high, the BDPCA + LDA would be more superior to the PCA + LDA in terms of the computational requirement. Since the FERET database is a much larger face database compared to the ORL database, the BDPCA + LDA would be much less computationally expensive than the PCA + LDA in the training phase.

V. CONCLUSION

In this correspondence, we propose a fast facial-feature-extraction technique, BDPCA + LDA, for face recognition. Two face databases, the ORL database and the FERET database, were used to evaluate the BDPCA + LDA. Experimental results show that the BDPCA + LDA framework has a number of significant advantages over the PCA + LDA framework. First of all, the BDPCA + LDA needs less computational requirement in both the training and the testing phases. The reason is twofold. On one hand, compared to the PCA + LDA, there are just some smaller eigenproblems required to be solved for the BDPCA + LDA. On the other hand, the BDPCA + LDA has a much faster speed for facial feature extraction. Second, the BDPCA + LDA needs less memory requirement because its projector is much smaller than that of the PCA + LDA. Third, the BDPCA + LDA has a higher recognition accuracy over the PCA + LDA.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and the editor for their constructive comments.

REFERENCES

[1] J. Baeka and M. Kimb, “Face recognition using partial least squares components,” *Pattern Recognit.*, vol. 37, no. 6, pp. 1303–1306, Jun. 2004.

[2] P. N. Belhumeour, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces versus Fisherfaces: Recognition using class specific linear projection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.

[3] H. Cevikalp, M. Neamtu, M. Wilkes, and A. Barkana, “Discriminative common vectors for face recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 1, pp. 4–13, Jan. 2005.

[4] R. Chellappa, C. L. Wilson, and S. Sirohey, “Human and machine recognition of faces: A survey,” *Proc. IEEE*, vol. 83, no. 5, pp. 705–740, May 1995.

[5] L. F. Chen, H. Y. M. Liao, M. T. Ko, J. C. Lin, and G. J. Yu, “A new LDA-based face recognition system which can solve the small sample size problem,” *Pattern Recognit.*, vol. 33, no. 10, pp. 1713–1726, Oct. 2000.

[6] S. Chen and Y. Zhu, “Subpattern-based principal component analysis,” *Pattern Recognit.*, vol. 37, no. 5, pp. 1081–1083, May 2004.

[7] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, “Indexing by latent semantic analysis,” *J. Amer. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391–407, 1990.

[8] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. New York: Academic, 1990.

[9] P. M. Garthwaite, “An interpretation of partial least squares,” *J. Amer. Stat. Assoc.*, vol. 89, no. 425, pp. 122–127, Mar. 1994.

[10] G. H. Golub and C. F. Van Loan, *Matrix Computation*, 3rd ed. Baltimore, MD: The Johns Hopkins Univ. Press, 1996.

[11] R. Gottumukkal and V. K. Asari, “An improved face recognition technique based on modular PCA approach,” *Pattern Recognit. Lett.*, vol. 25, no. 4, pp. 429–436, Mar. 2004.

[12] A. Habibi and P. A. Wintz, “Image coding by linear transformation and block quantization,” *IEEE Trans. Commun.*, vol. COMM-19, no. 1, pp. 50–62, Feb. 1971.

[13] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[14] M. Kirby and L. Sirovich, “Application of the KL procedure for the characterization of human faces,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 1, pp. 103–108, Jan. 1990.

[15] W. Liu, Y. Wang, S. Z. Li, and T. Tan, “Null space approach of Fisher discriminant analysis for face recognition,” in *Proc. BioAW*, D. Maltoni and K. Jain, Eds., Prague, Czechoslovakia, 2004, vol. 3087, Lecture Notes in Computer Science, pp. 32–44.

[16] C. Liu and H. Wechsler, “Enhanced Fisher linear discriminant models for face recognition,” in *Proc. 14th Int. Conf. Pattern Recognition*, Brisbane, Australia, Aug. 1998, vol. 2, pp. 1368–1372.

[17] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, “Face recognition using LDA-based algorithms,” *IEEE Trans. Neural Netw.*, vol. 14, no. 1, pp. 195–200, Jan. 2003.

[18] *The ORL Face Database*, Cambridge, U.K.: AT&T (Olivetti) Research Laboratories. [Online]. Available: http://www.uk.research.att.com/facedatabase.html.

[19] P. J. Philips, *The Facial Recognition Technology (FERET) Database*. [Online]. Available: http://www.itl.nist.gov/iad /humanid/feret.

[20] P. J. Philips, H. Moon, S. A. Rizvi, and P. J. Rauss, “The FERET evaluation methodology for face-recognition algorithms,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1090–1104, Oct. 2000.

[21] W. K. Pratt, *Digital Image Processing*, 3rd ed. New York: Wiley, 2001.

[22] W. D. Ray and R. M. Driver, “Further decomposition of the Karhunen-Loeve series representation of a stationary random process,” *IEEE Trans. Inf. Theory*, vol. IT-16, no. 6, pp. 663–668, Nov. 1970.

[23] L. Sirovich and M. Kirby, “Low-dimensional procedure for characterization of human face,” *J. Opt. Soc. Amer.*, vol. 4, no. 3, pp. 519–524, Mar. 1987.

[24] D. L. Swets and J. Weng, “Using discriminant eigenfeatures for image retrieval,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 8, pp. 831–836, Aug. 1996.

[25] K. Torkkola, “Linear discriminant analysis in document classification,” in *Proc. IEEE ICDM Workshop Text Mining*, San Jose, CA, 2001. [Online]. Available: http://www-ai.ijs/DunjaMladenic/TextDM01

[26] M. Turk and A. Pentland, “Eigenfaces for recognition,” *J. Cogn. Neurosci.*, vol. 3, no. 1, pp. 71–86, 1991.

[27] J. Yang and J. Y. Yang, “From image vector to matrix: A straightforward image projection technique—IMPCA vs. PCA,” *Pattern Recognit.*, vol. 35, no. 9, pp. 1997–1999, Sep. 2002.

[28] ——, “Why can LDA be performed in PCA transformed space,” *Pattern Recognit.*, vol. 36, no. 2, pp. 563–566, Feb. 2003.

[29] J. Yang, J. Y. Yang, and A. F. Frangi, “Combined Fisherfaces framework,” *Image Vis. Comput.*, vol. 21, no. 12, pp. 1037–1044, Nov. 2003.

[30] J. Yang, D. Zhang, A. F. Frangi, and J.-Y. Yang, “Two-dimensional PCA: A new approach to appearance-based face representation and recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 131–137, Jan. 2004.

[31] H. Yu and J. Yang, “A direct LDA algorithm for high-dimensional data with application to face recognition,” *Pattern Recognit.*, vol. 34, no. 10, pp. 2067–2070, Oct. 2001.

[32] W. Zhao, R. Chellappa, and P. J. Philips, “Subspace linear discriminant analysis for face recognition,” Center Autom. Res., Univ. Maryland, College Park, Tech. Rep. CAR-TR-914, 1999.

[33] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, “Face recognition: A literature survey,” *ACM Comput. Surv.*, vol. 35, no. 4, pp. 399–458, Dec. 2003.

[34] W. Zheng, L. Zhao, and C. Zou, “An efficient algorithm to solve the small sample size problem for LDA,” *Pattern Recognit.*, vol. 37, no. 5, pp. 1077–1079, May 2004.