Available online at www.sciencedirect.com

ScienceDirect

Computer Vision and Image Understanding 110 (2008) 91–101

Computer Vision and Image Understanding

www.elsevier.com/locate/cviu

Two-stage optimal component analysis

Yiming Wu [a,*], Xiuwen Liu [a], Washington Mio [b], K.A. Gallivan [c]

[a] *Department of Computer Science, Florida State University, Tallahassee, FL 32306, USA*
[b] *Department of Mathematics, Florida State University, Tallahassee, FL 32306, USA*
[c] *School of Computational Science, Florida State University, Tallahassee, FL 32306, USA*

Received 26 September 2006; accepted 30 April 2007
Available online 8 June 2007

Abstract

Linear techniques are widely used to reduce the dimension of image representation spaces in applications such as image indexing and object recognition. Optimal Component Analysis (OCA) is a method that addresses the problem of learning an optimal linear representation for a particular classification task. The problem is formulated in the framework of optimization on a Grassmann manifold and treated with stochastic gradient methods intrinsic to the manifold. OCA has been successfully applied to image classification problems arising in a variety of contexts. However, as the search space is typically very high dimensional, OCA optimization often requires a large number of iterations, each involving extensive computations that make the algorithm somewhat costly to implement. In this paper, we propose a two-stage method, which we refer to as two-stage OCA, that improves the search efficiency by orders of magnitude without compromising the quality of the estimation. In fact, extensive experiments using face and object classification datasets indicate that the proposed method often leads to more accurate classification than the original OCA since it is not as prone to over-fitting. Two-stage OCA also leads to substantial improvement in classification performance as compared to other linear dimension reduction methods.

Keywords: Linear representation; Object recognition; Optimal component analysis; Grassmann manifold

1. Introduction

The wide availability of affordable imaging devices such as digital cameras and camcorders has reinvigorated the interest of computer vision researchers in image-based object detection and recognition for a broad spectrum of applications including video surveillance and the development of intelligent human–computer interfaces. As image representations tend to be very high dimensional, dimension reduction methods are of central importance for robust inference and computational efficiency. It is well-known that variations observed in images representing a specific class of objects, such as human faces, are typically constrained to (possibly nonlinear) subspaces of much smaller dimension. These variations are due to differences in the physical attributes of the subjects as well as to a number of imaging parameters such lighting, orientation and camera distance. While nonlinearity is inherent to imaging processes and reflectance properties, linear representations often provide satisfactory approximations for tasks such as indexing and recognition (cf. [1,2]). In the context of image classification, the primary goal of linear dimension reduction is to obtain linear low-dimensional approximations that capture and retain only the most relevant features for the particular classification task at hand.

To be specific, let I be an image reshaped into $n \times 1$ vector and let U be a $n \times d$ matrix, whose columns form an orthonormal basis of a d-dimensional subspace S of $\Re^n$. The vector $\alpha(I) = U^{\mathrm{T}} I \in \Re^d$, known as the vector of image coefficients, provides a d-dimensional representation of the orthogonal projection of I onto the subspace S. Recognition and other processes can then be based on $\alpha(I)$ instead of I, resulting in substantial complexity reduction in the representation and more efficient algorithms, if $d \ll n$.

[*] Corresponding author. Fax: +1 850 644 0058.
E-mail address: ywu@cs.fsu.edu (Y. Wu).

1077-3142/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2007.04.005

The main goal of this paper is to investigate dimension-reduction techniques that are robust and effective for image-based object recognition. Commonly used linear methods include Principal Component Analysis (PCA) (e.g. [3–5]), Linear Discriminant Analysis (LDA) (e.g. [6–8]), and Independent Component Analysis (ICA) (e.g. [9,10]). PCA is an unsupervised method which makes use of the data covariance to estimate the principal modes of variation; orthogonal projections onto the principal directions yield variables that are uncorrelated. PCA optimally retains the variance of the data in the reduced feature space and minimizes the reconstruction error for the training images. However, in general, PCA does not lead to an optimal solution as transformations that maximize the variance are not necessarily optimal for recognition [6]. Note that PCA does not utilize statistics higher than second order. Higher order statistics can be accounted for with ICA, which estimates statistically independent components. However, ICA is not designed to optimize recognition, either. LDA, on the other hand, aims to find an optimal basis that separates the means of different classes as much as possible. If the underlying distributions of all the classes are Gaussian with the same variance, it can be shown that LDA achieves maximum discrimination [6]. Note that the optimality is not valid when the underlying models do not satisfy this assumption (e.g. [11,12]). The basis given by classical LDA can be readily computed by solving a generalized eigen-decomposition involving scatter matrices, which requires one of the scatter matrices to be nonsingular. However, in many real applications, such as face recognition and text classification, all scatter matrices in question can be singular since the dimension of data, in general, by far exceeds the number of data points—this is known as the *singularity* problem [13,14]. In order to deal with the singularity issue in under-sampled problems, several extended LDA methods have been proposed recently for recognition applications, such as pseudo-inverse LDA [15], regularized LDA [16], PCA + LDA [17,18], LDA/GSVD [19], and LDA/QR [20].

As distributions of real images are typically non-Gaussian (cf. [21]), for image classification and object recognition applications, there is no theoretical basis for the most common choices of dimension reduction methods; this is also evident from (often contradictory) comparative studies reported in the literature (e.g. [11,12,17]). In fact, one can construct simple scenarios in which all standard choices yield the worst classification performance. Such an example is shown in Fig. 1, which consists of two classes ('+' and '×') with eight points each, and the points are present in clusters of fours. It is easy to show the one-dimensional subspace resulting from PCA, ICA, and FDA coincides with either the horizontal or the vertical axis. If we use the nearest neighbor classifier and let a point from each cluster be used for training, the one-dimensional PCA, ICA, and FDA basis gives the worst performance. However, many other subspaces, such as the one shown in Fig. 1(c) provides optimal performance. This provides compelling evidence that, in the context of classification and recognition, a more relevant question is to find an optimal linear representation tailored to a particular classification problem. Unlike the commonly used dimension reduction methods, Optimal Component Analysis (OCA) [22,23] is application specific and has been applied successfully to many pattern recognition problems. The search for an optimal subspace is based on a stochastic gradient process that seeks to maximize a specified performance function over all subspaces of a particular dimension; that is, over the elements of a Grassmann manifold. A solution is obtained by conducting a search over the Grassmannian with an MCMC (Markov chain Monto Carlo) type algorithm. OCA provides a computational framework for finding optimal linear representations for particular applications and its effectiveness has been demonstrated on many real datasets [22].

The main limitations of OCA are the number of iterations required in a typical search and the computational costs associated with the estimation of gradients due to the high dimensionality of the data. In this paper, we propose a two-stage strategy that improves the search efficiency, overcomes the computational problems of OCA, and better addresses issues related to over-fitting. (A preliminary, short version of this work appeared in [24].) In the first stage, we obtain a more compact representation of the input images by dimension reduction with a computationally efficient method, such as PCA, ICA, LDA, Random Component Analysis [25], or QR decomposition. Then, in the second stage, OCA is employed to further reduce the dimension, but this time also optimizing the classification performance. As the dimension of the OCA search space is now much lower, an optimal basis can be obtained more efficiently. We show empirically that we can often reduce the dimension to a (much) smaller dimension and still achieve high classification performance.

The rest of the paper is organized as follows. Section 2 provides a brief review of OCA and the proposed two-stage OCA method is presented in Section 3. Several experimental results using two-stage OCA on face and object datasets are reported in Section 4. We conclude the paper with a summary and a discussion of future work in Section 5.

2. Optimal component analysis

Optimal component analysis is a dimension reduction technique that finds an optimal subspace (of a prescribed dimension) of feature space that optimizes the ability of the nearest neighbor classifier to index and classify images or more general data. The measurement of optimality is based on training data and the algorithm yields an orthonormal basis of the estimated optimal subspace.

More specifically, let $U \in \Re^{n \times d}$ be a matrix whose columns form an orthonormal basis of a d-dimensional subspace of $\Re^n$, where n is the size of the input image and d is the dimension of the desired subspace (generally $n \gg d$). For an image I, considered as a column vector of

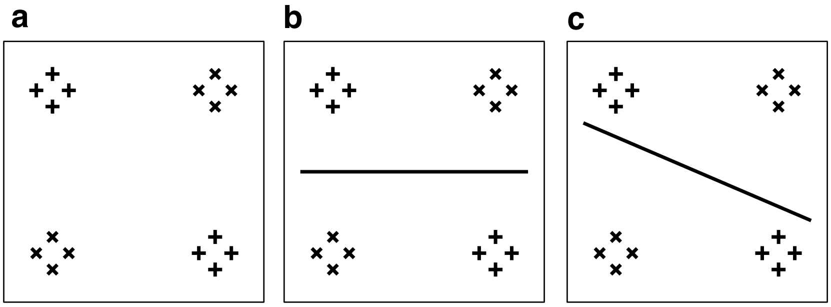

Fig. 1. A synthetic dataset consisting of two classes, each consisting of two clusters with four points. (a) The data set of two classes ('+' and '×') with eight points each in $\mathbb{R}^2$. (b) The one-dimensional subspace from PCA, ICA, and FDA. (c) A one-dimensional subspace optimal for recognition using the nearest neighbor classifier.

size n, the vector of coefficients is given by $\alpha(I,U) = U^{\mathrm{T}}I \in \Re^d$ and represents the orthogonal projection of I onto the subspace S_U spanned by the columns of U. Suppose the training data consists of representatives of C classes of images, with each class represented by k_{train} training images (denoted by $I_{c,1},\ldots,I_{c,k_{\text{train}}}$) and k_{cross} cross validation images (denoted by $I'_{c,1},\ldots,I'_{c,k_{\text{cross}}}$). Let

$$\rho(I'_{c,i},U) = \frac{\min\limits_{c'\neq c,j} D(I'_{c,i},I_{c',j};U)}{\min\limits_{j} D(I'_{c,i},I_{c,j};U)+\epsilon}. \quad (1)$$

The numerator is the distance from $I'_{c,i}$ to the closest training image not in its class and the denominator is the distance from $I'_{c,i}$ to the closest training image in the same class. Here, D denotes Euclidean distance; that is,

$$D(I_1,I_2;U) = \|\alpha(I_1,U)-\alpha(I_2,U)\|, \quad (2)$$

where $\|\cdot\|$ is the usual 2-norm. In Eq. (1), $\epsilon > 0$ is a small number introduced to avoid division by zero. Note that large values of ρ are desirable, since this means that $I'_{c,i}$ will be closer to its class than to other classes in the subspace S_U. A performance function F is defined to essentially measure the average value of ρ over all cross-validation images, as follows:

$$F(U) = \frac{1}{Ck_{\text{cross}}}\sum_{c=1}^{C}\sum_{i=1}^{k_{\text{cross}}} h(\rho(I'_{c,i},U)-1), \quad (3)$$

where $h(\cdot)$ is a monotonically increasing bounded function used to control bias with respect to particular classes in measurements of performance. In our implementation, we use $h(x) = 1/(1+\exp(-2\beta\, x))$, where β is a parameter that controls the degree of smoothness of $F(U)$. Thus, F is a quantifier of the ability of the nearest neighbor classifier to discern the C classes after projection onto S_U. Moreover, as $\beta \to \infty$ and $\epsilon \to 0$, F gives precisely the recognition performance of the nearest neighbor classifier after projection to the subspace given by U [22].

Under this formulation, $F(U) = F(UH)$ for any $d\times d$ orthogonal matrix H. This is the case because F depends only on distances in S_U and right multiplication by H changes the orthonormal basis, but not the subspace S_U. Therefore, our search for optimal representation can be viewed as an optimization problem on the space of d-dimensional subspaces rather than the space of orthonormal frames. The Grassmann manifold, $\mathscr{G}(n,d)$, is the set of all d-dimensional subspaces of $\Re^n$ [26]. It is a compact, connected manifold of dimension $d(n-d)$, which can be represented either by a basis (non-uniquely) or by a projection matrix (uniquely). Choosing the former, let U be an $n\times d$ matrix whose columns are an orthonormal basis for the given subspace of $\Re^n$ and let $[U]$ denote the set of all the orthonormal bases of S_U, i.e., $[U] = \{UH | H \in \Re^{d\times d}, H^{\mathrm{T}}H = I_d\} \in \mathscr{G}(n,d)$. The remarks above imply that F is a function of $[U]$, not just U. Unlike the actual recognition performance, $F([U])$ is smooth and thus allows us to use a gradient-type algorithm to solve the optimization problem. An optimal d-dimensional subspace for the given classification problem from the viewpoint of the available data is given by

$$[\hat{U}] = \underset{[U]\in\mathscr{G}_{n,d}}{\operatorname{argmax}} F([U]). \quad (4)$$

In [22], an optimization algorithm utilizing the geometric properties of the manifold is presented. A Monte Carlo version of a stochastic gradient-based algorithm with simulated annealing is used to find an optimal subspace Û. Since the gradient search is conducted over a Grassmann manifold, the process has to account for its intrinsic geometry. In [26], the Newton–Raphson method has been studied on such manifolds. We now review the MCMC-type simulated annealing process presented in [22].

Let J be the $n\times d$ matrix given by the first d columns of the $n\times n$ identity matrix. Complete the orthonormal set U to an orthonormal basis of $\Re^n$ and let Q be the corresponding $n\times n$ orthogonal matrix. Then, the gradient of F at $[U]$ is given by $A([U])J$, where

$$A([U]) = Q\sum_{i=1}^{d}\sum_{j=d+1}^{n}\alpha_{ij}(U)E_{ij} \in \Re^{n\times n},$$

where

$$\alpha_{ij}(U) = \lim_{\epsilon\to 0}\frac{F(Qe^{\epsilon E_{ij}}J)-F(U)}{\epsilon}$$

is the directional derivatives of F in the directions given by E_{ij}. Here E_{ij} is an $n \times n$ skew-symmetric matrix

$$E_{ij}(k,l) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } k=i,\ l=j, \\ -\frac{1}{\sqrt{2}} & \text{if } k=j,\ l=i, \\ 0 & \text{otherwise}, \end{cases} \tag{5}$$

$1 \leqslant i \leqslant d$ and $d < j \leqslant n$. The matrices $E_{ij}J$ represent an orthonormal basis of the vector space tangent to $\mathcal{G}(n,d)$ at $[J]$. The deterministic gradient flow is a solution of the equation

$$\frac{\mathrm{d}U_t}{\mathrm{d}t} = A(U_t)J, \tag{6}$$

where U_t is the solution at time t. Computationally, we discretely update U_t according Eq. (6) and at each iteration, the gradient vector of F with respect to U_t is computed. This gives rise to a deterministic gradient optimization algorithm that is intrinsic to the Grassmann manifold, i.e., every new solution is guaranteed to be on the manifold given that U_0 is. This algorithm shares the limitations of all deterministic gradient algorithms and it will not be able to escape a local maximum. To overcome this problem, in [22] stochastic optimization is used by first perturbing the gradient randomly and then using a Markov chain Monte Carlo process. A proposed subspace is accepted with a probability that depends on the performance improvement and an annealing parameter. If the performance on the new subspace is better than that of the current solution, it is always accepted; otherwise, the worse the performance, the lower the probability of the subspace being accepted. This guarantees that a global optimal solution[1] can be reached given that the Markov chain is sufficiently long. For details, see [22].

The computational complexity of each iteration of the algorithm is $C_n = O(d \times (n-d) \times k_{\text{cross}} \times k_{\text{training}} \times n \times d)$. C_n is obtained by the following observations. The dimension of the gradient vector is $d \times (n-d)$, which can be seen from Eq. (5) (as there are $d \times (n-d)$ E_{ij}'s). For each E_{ij}, in order to compute $\alpha_{ij}(U)$, we need to compute $F(e^{\epsilon E_{ij}}U)$, which requires to compute the ratio (Eq. (1)) for each cross validation image, which again requires a search of the closest training image in the same class and the closest in other classes. Therefore estimating the gradient requires the given computational complexity. By exploiting the structure of $A(U)$, an $O(n)$ updating algorithm can be achieved and thus it can be ignored. The overall computational complexity is therefore $C_n \times T$, where T is the number of iterations.

Note that the OCA algorithm requires solving an optimization problem with dimension of $d \times (n-d)$; for recognition applications based on images, n is typically on the order of 10,000. In [22], OCA has been implemented and demonstrated on recognition problems with n of 10,000. However, due to its computational complexity, OCA has not been used widely for recognition applications.

3. Two-stage OCA

A significant reduction of the computational complexity can be achieved by restricting the OCA search to d-dimensional subspaces of the span of the training images. If the dataset contains N images, $I_1,\ldots, I_N$, we arrange them as $D_N = [I_1, I_2, \ldots, I_N] \in \Re^{n \times N}$. If the rank of D_N is r, let D be an $n \times r$ matrix such $D^{\mathrm{T}}D = I_r$ and whose columns form a basis of the span of the training set. Then, $D^{\mathrm{T}}I \in \Re^r$ gives a reduced representation of an image $I \in \Re^n$. In typical recognition problems based on images, $r \ll n$, so that the OCA search can be carried out much more efficiently in this r-dimensional representation as the dimension of the Grassmann manifold is reduced from $d(n-d)$ to $d(r-d)$. Note that, in this type of preliminary dimension reduction, all the information contained in the original training set is retained. According to the analysis given in the last section, the computation complexity of each OCA iteration is then $O(r^2)$ instead of $O(n^2)$.

This gives rise to a two-stage OCA algorithm. Instead of solving the OCA optimization in the original image space, we limit the search to the span of the training images using a lower dimensional representation. To achieve even higher efficiency, in practice, we may want to further reduce the dimension using a computationally efficient dimension reduction method first. We refer to this step as pre-dimension reduction.

An immediate question is how to choose pre-dimension reduction technique. Note that the performance is essentially determined by the distance between images in the reduced space (see Eqs. (2) and (3)), therefore, any method that retains the effective discriminative subspace would be sufficient. Two choices seem to be most relevant. First, we can choose to minimize the average reconstruction error, which can be achieved using PCA [4,7]. An alternative is to choose the components that are most discriminative assuming the underlying distributions are Gaussian with fixed variance; this can be achieved using FDA by solving a generalized eigenvalue problem [6]. However, as pointed out earlier, there is no theoretical basis for choosing PCA or FDA, in general.

In addition, for very large datasets, more efficient methods may be preferred as both PCA and FDA require solving an eigenvalue problem. In particular, to be more scalable, Ye and Li [20] proposed a two-stage QR/LDA method. In the first stage, QR is applied to the matrix consisting of the mean image of each class and therefore a much small matrix (compared to the covariance matrix of the entire dataset); additionally QR can be solved more efficiently as the ordering of different dimensions is not required. Then LDA is applied in the reduced space, which, in addition to the efficiency, also avoids the singularity problem; see [20] for details. Here we use the QR decompo-

[1] Note that the solution of Eq. (4) can be a set rather than a unique subspace.

sition in the first stage of our QR/OCA algorithm to achieve the scalability and efficiency.

To summarize, our two-stage OCA method is implemented as follows: in the first stage, we reduce the input data from the original high dimension to a lower dimension using a computationally efficient method; in the second stage, an OCA search is performed in the reduced space. As the search space is (much) smaller than the original one, the computational cost is greatly reduced. The computational costs for OCA on two Grassmann manifolds $G_{n,d}$ and $G_{n_1,d}$ (where $n_1 = n/m$, m is the dimension reduction factor, $m \gg 1$, and $n \gg d$, and $n_1 > d$) can be compared. For each iteration, the computational complexity with images of size n is $C_n = O(d \times (n - d) \times k_{\text{cross}} \times k_{\text{training}} \times n \times d)$. For $n_1 = n/m$, the computational complexity with images of size n_1 is

$$\begin{aligned} C_{n_1} &= O\left(d \times \left(\frac{n}{m} - d\right) \times k_{\text{test}} \times k_{\text{training}} \times \frac{n}{m} \times d\right) \\ &= \frac{n - md}{m^2(n - d)} C_{N_1} \\ &\approx \frac{1}{m^2} C_{n_0}, \end{aligned}$$

considering the fact $n \gg d$. Obviously it is more efficient to learn on $\mathcal{G}_{n_1,d}$ than on $\mathcal{G}_{n,d}$ for the dimension of search space is reduced from $d \times (n - d)$ to $d \times (n_1 - d)$. Additionally, since the search in the reduced space is more effective, the number of required iterations can be much smaller, demonstrated by the experimental results shown in Fig. 3.

4. Experimental results

We evaluate numerically the effectiveness of the two-stage OCA algorithm in this section. The data sets used for performance study are summarized in Section 4.1. In Section 4.2 we compare the efficiency and effectiveness of the proposed two-stage OCA algorithm to that of the original OCA algorithm. Then, in Section 4.3, we present the classification accuracy of two-stage OCA algorithm using different dimensional reduction methods in the first stage of the algorithm, such as PCA/OCA, ICA/OCA, LDA/OCA, RCA/OCA, and QR/OCA; we also compare the PCA/OCA algorithm with other classification methods, such as PCA, PCA + LDA, and LDA/QR. The program is implemented in C and the experiments are conducted on a workstation with an Intel Xeon 3.00 GHz CPU with 8GB of RAM.

4.1. Data sets

We have used four data sets for performance evaluations and comparisons, including three well-known face datasets and one 3D object recognition dataset. Fig. 2 gives some sample images of these data sets.

- ORL face data set [27]. It contains 400 centralized face images of 40 individuals with image size of 92 × 112. The major challenge on this data set is the variation of the face poses as there is no lighting variation with min-

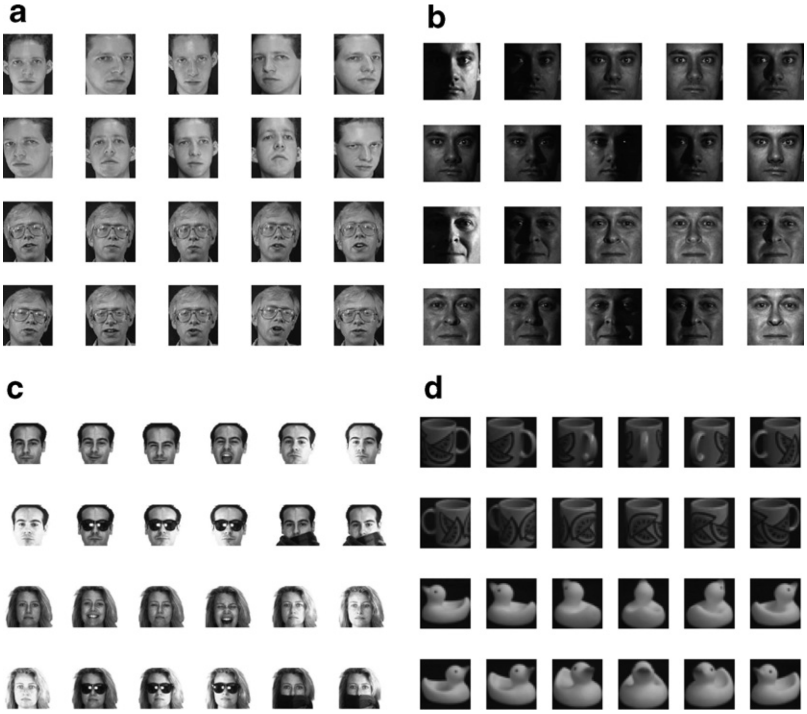

Fig. 2. Some example images of data sets used in experiments. (a) *ORL*; (b) *PIE*; (c) *AR*; (d) *COIL*.

imal facial expression variation and no occlusion. We use the whole images as the input, i.e., the dimension of an instance is $92 \times 112 = 10{,}304$.

- PIE face data set [28]. We only use part of the whole data set, where cropped face images are available. The subset used in our experiments contains 66 people with 21 images each. The image size is $100 \times 100 = 10{,}000$.
- AR face data set [29]. It is a large face image data set. The images contain significantly large areas of occlusion, due to the presence of sun glasses and scarves. The existence of occlusion dramatically increases the within-class variations of AR face image data and makes the recognition more difficult. In order to compare with results shown in [14], here we use the same subset of AR, containing 1638 face images of 126 individuals and each image is pre-processed the same way as in [14]. Specifically, as used in [14], we first crop the original images (of size 768×576) from rows 100 to 500 and columns 200–550, and then subsample the cropped images with sample step 4×4. The dimension of each image is reduced to $101 \times 88 = 8888$ as in [14].
- COIL object data set [30]. The COIL-100 data set consists of images of 100 3D objects with 72 images (of different poses) per object. The images in the data set are normalized such that the larger of the two object dimensions (height and width) fits a 32×32 area.

4.2. Comparison with original OCA

In this set of experiments, we study the computational efficiency gain of the proposed two-stage OCA algorithm by comparing its running time to that of the original OCA algorithm. Table 1 summarizes the data sets used in the experiments. Table 2 shows the running time, recognition ratio F and classification accuracies with respect to the dimension kept in the first stage of PCA/OCA on the four data sets. It shows clearly that the two stage OCA algorithm speeds up the original OCA algorithm dramatically. For example, on the *ORL* data set, the running time of original OCA is about 2 days (for 1000 iterations), while it only takes 971 s when we reduce the dimension from 10,304 to 50. In addition to the significant time reduction, the two-stage OCA also gives a solution that leads to a higher classification accuracy. On the *AR* data set, the accuracy improvement is significant, from 92% given by the original OCA to 97% given by PCA/OCA with the dimension reduced to 500 in the first stage.

Table 1
Statistics for our real test data sets

Dataset	Dimension	Classes	Number of images per class		
			Total	Training	Test
ORL	10,304	40	10	5	5
PIE	10,000	66	21	10	11
AR	8888	126	13	6	7
COIL	1024	100	72	36	36

Table 2
Time (in seconds per iteration), recognition ratio F and classification accuracy (%) of PCA/OCA

ORL	Dimension	10,304	199	150	100	50	20
	Time	173	10.00	7.20	2.10	0.97	0.67
	F	0.9523	0.99340	0.9928	0.9967	0.9985	0.9969
	Accuracy	100	100	100	100	100	100
PIE	Dimension	10,000	599	300	100	50	20
	Time	224	15.20	8.42	2.24	1.27	0.70
	F	0.9534	0.9921	0.9924	0.9967	0.9987	0.9938
	Accuracy	98.76	98.21	98.07	100	99.45	97.66
AR	Dimension	8888	500	300	100	50	20
	Time	421	15.20	8.42	1.13	1.07	0.54
	F	0.9245	0.9779	0.9761	0.9675	0.9467	0.9302
	Accuracy	92.12	97.01	93.66	95.80	93.66	92.12
COIL	Dimension	1,024	500	300	100	50	20
	Time	531	32.20	9.42	1.53	1.34	0.78
	F	0.9482	0.9504	0.9587	0.9491	0.9457	0.9424
	Accuracy	98.78	97.86	98.70	99.28	97.94	94.36

As shown in the last section, when the dimension is reduced, it will also be more efficient to learn an optimal basis in the lower dimensional space. Fig. 3 shows the evolution of performance ratio F of the original OCA and PCA/OCA on *ORL* and *PIE* data sets. For example, Fig. 3(a) shows the evolution of F of the original OCA algorithm and PCA/OCA algorithms with dimension reduced to 19 and 50 in the first stage. As the plots show, not only PCA/OCA achieves higher performance ratio than the original OCA, and it also takes much fewer iterations to reach nearly 1.0 in the reduced space. While in the original space, the performance ratio is much lower and is not close to 1.0 after 1000 iterations. The results on the *PIE* data set, shown in Fig. 3(b) are also consistent, where PCA/OCA algorithms achieve the best performance of the original OCA algorithm (in 1000 iterations) in less than 20 iterations. Besides PCA/OCA, other two-stage algorithms show similar patterns, where the results using LDA/OCA is shown in Fig. 4 for *ORL* and *PIE* data sets. These results, along with that shown in Table 2, clearly demonstrate the significant improvement in efficiency and effectiveness of the proposed two-stage OCA method compared to the original OCA.

4.3. Classification accuracy

As indicated in Section 3, different dimensional reduction methods can be used in the first stage of the two stage OCA, such as PCA, IDA, LDA RCA, QR, and etc. In the following set of experiments, we evaluate the two-stage OCA algorithm in terms of classification accuracy on *ORL*, *PIE*, *AR* and *COIL* datasets. For all the datasets, we reduce the dimension from the original dimension to 100 in the first stage. Table 3 shows the classification accuracy with respect to different dimensional reduction meth-

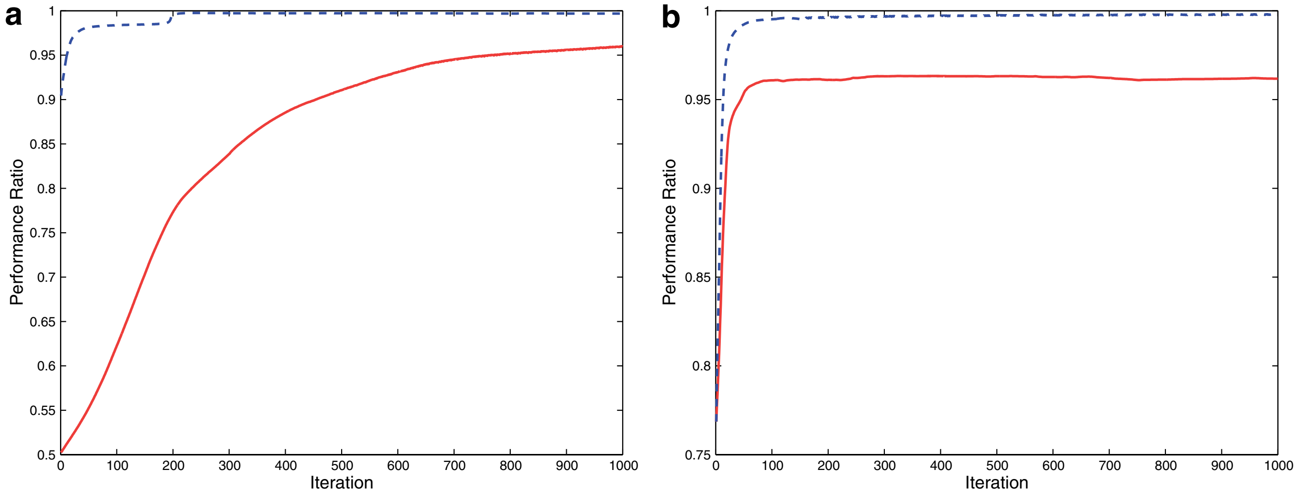

Fig. 3. Evolution of performance function *F* versus iteration number *t* of the original OCA and PCA/OCA algorithm on *ORL* and *PIE* data sets. (a) For *ORL* data set. Solid line (red): the original OCA where $n_0 = 10304$; dashed line (blue): PCA/OCA where $n_1 = 19$. (b) For *PIE* data set. Solid line (red): the original OCA where $n_0 = 10{,}000$; dashed line (blue): PCA/OCA where $n_1 = 50$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

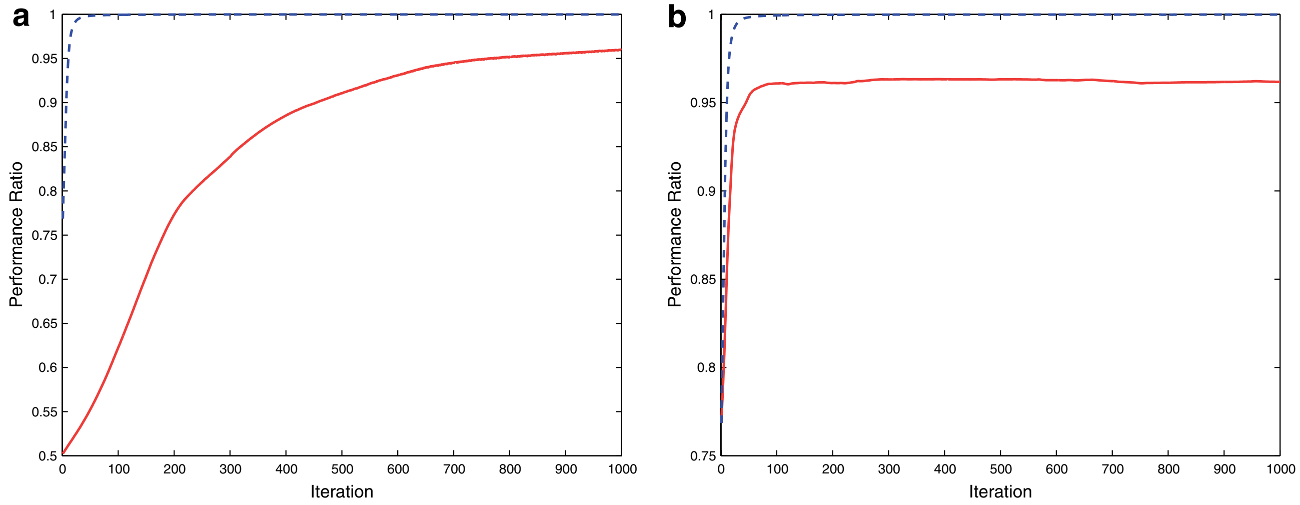

Fig. 4. Evolution of performance function *F* versus iteration number *t* of the original OCA and LDA/OCA on *ORL* and *PIE* data sets. (a) For *ORL* data set. Solid line (red): the original OCA where $n_0 = 10{,}304$; dashed line (blue): LDA/OCA where $n_1 = 19$. (b) For *PIE* data set. Solid line (red): the original OCA where $n_0 = 10{,}000$; dashed line (blue): LDA/OCA where $n_1 = 50$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

ods used in the first stage. From these results, we have the following important observations:

- All the two-stage OCA methods achieve high classification accuracy on these datasets, especially for *ORL* and *PIE* datasets.
- LDA/OCA achieves best classification accuracy on *PIE* and *AR* datasets. These may indicate the underlying distributions in these data sets can be approximated well by Gaussian distributions in the reduced space and LDA preserves most of the discriminative dimensions of the data. Note that LDA or variant LDAs in general do not perform as well (see Table 4).
- PCA/OCA achieves best classification accuracy on the *COIL* data set except one case.
- As the *AR* and *COIL* datasets are more challenging, the classification accuracy is worse than that on the *ORL* and *PIE* datasets; note however, compared to other methods shown in Table 4, two-stage OCA algorithms give marked improvement in accuracy on the AR dataset.

Fig. 5 shows the evolution of performance and classification accuracy of the proposed PCA/OCA algorithm on *ORL*, *PIE*, *AR* and *COIL* data sets. In each panel, the upper plot in shows the evolution of classification accuracy

Table 3
Classification accuracy (%) of two-stage OCA on the four data sets

Data set	KNN	PCA/OCA	ICA/OCA	RCA/OCA	LDA/OCA	QR/OCA
ORL	1	100	100	97.5	100	100
	3	100	100	95.0	100	100
	4	100	97.5	90.0	100	100
	5	100	97.5	90.0	100	100
	10	100	95.0	87.5	100	100
PIE	1	100	98.07	93.66	100	100
	3	99.45	95.32	81.82	100	98.76
	4	98.76	93.66	80.44	100	98.21
	5	98.21	99.11	79.20	100	98.07
	10	97.66	98.21	73.56	100	93.66
AR	1	95.80	92.45	92.06	99.89	93.66
	3	92.45	93.66	80.50	99.77	92.45
	4	92.12	92.12	80.16	99.54	92.12
	5	90.63	90.02	79.37	99.43	93.66
	10	85.63	84.36	76.98	97.01	90.63
COIL	1	99.28	97.86	96.13	98.78	97.22
	3	98.70	94.36	92.50	97.94	94.36
	4	97.94	92.58	88.95	96.83	91.66
	5	94.75	92.52	88.22	95.33	92.00
	10	92.95	86.58	84.26	92.50	86.81

Table 4
The classification accuracy (%) of different dimension reduction methods on *ORL* and *AR* data set

Data set	KNN	PCA	PCA + LDA	LDA/GSVD	QR/LDA
ORL	1	97.25(1.42)	95.00(3.12)	94.00(3.76)	98.25(1.69)
	3	94.50(3.07)	94.75(3.43)	94.00(3.76)	98.00(2.58)
	5	92.25(2.99)	95.50(2.58)	94.00(3.76)	98.25(2.06)
	10	81.25(5.03)	93.75(3.58)	94.00(3.76)	96.75(2.37)
AR	1	65.30(2.63)	92.45(1.22)	92.60(1.16)	98.41(1.26)
	3	59.05(2.10)	90.72(1.17)	92.60(1.16)	94.03(1.63)
	5	57.49(2.04)	88.50(1.17)	92.60(1.16)	89.53(1.67)
	10	44.70(2.49)	85.63(1.84)	92.60(1.16)	86.93(2.44)
	KNN	OCA	PCA/OCA	QR/OCA	LDA/OCA
ORL	1	100(−)	100(0)	100(0)	100(0)
	3	100(−)	100(0)	100(0)	100(0)
	5	100(−)	100(0)	100(0)	100(0)
	10	100(−)	100(0)	100(0)	100(0)
AR	1	100(−)	99.21(1.32)	98.35(1.23)	99.87(0.85)
	3	96.49(−)	94.78(2.45)	98.21(1.21)	96.03(1.08)
	5	95.24(−)	94.11(1.32)	93.87(2.84)	95.86(0.94)
	10	92.74(−)	88.10(3.65)	87.10(3.21)	92.13(2.56)

while the lower plot shows the evolution of performance function. These plots show that the performance achieves 100% after less than 200 iterations for *ORL* data set, for *PIE* data set, it can achieve 100% for less than 10 iterations; for *AR* and *COIL* data sets, the classification accuracy is a little worse, but the proposed method achieves over 95% in 500 iterations. As shown in the lower plots, for *ORL* and *PIE* face data sets, the performance ratio F is close to 1, indicating all the images are classified correctly and robustly. For the *AR* and *COIL* data sets, the performance ratio is also high, indicating the classification is robust. Fig. 6 shows the evolution of performance and classification accuracy of the LDA/OCA algorithm on the data sets and the plots also demonstrate the effectiveness of two-stage OCA.

The next set of experiments compares the performance of two-stage OCA methods with other classification methods, including PCA, PCA + LDA, LDA/GSVD, QR/LDA, and the original OCA algorithm. The *ORL* and *AR* face data sets are used in these experiments. For a fair comparison, we use the results of other methods from [20]. In [20], the relevant parameters are as follows: 100 principal components of PCA are used in the PCA stages of PCA + LDA. For LDA algorithm, the output dimension is $k - 1$, where k is the number of classes, as the k centroids in all data sets are linearly independent. The classification

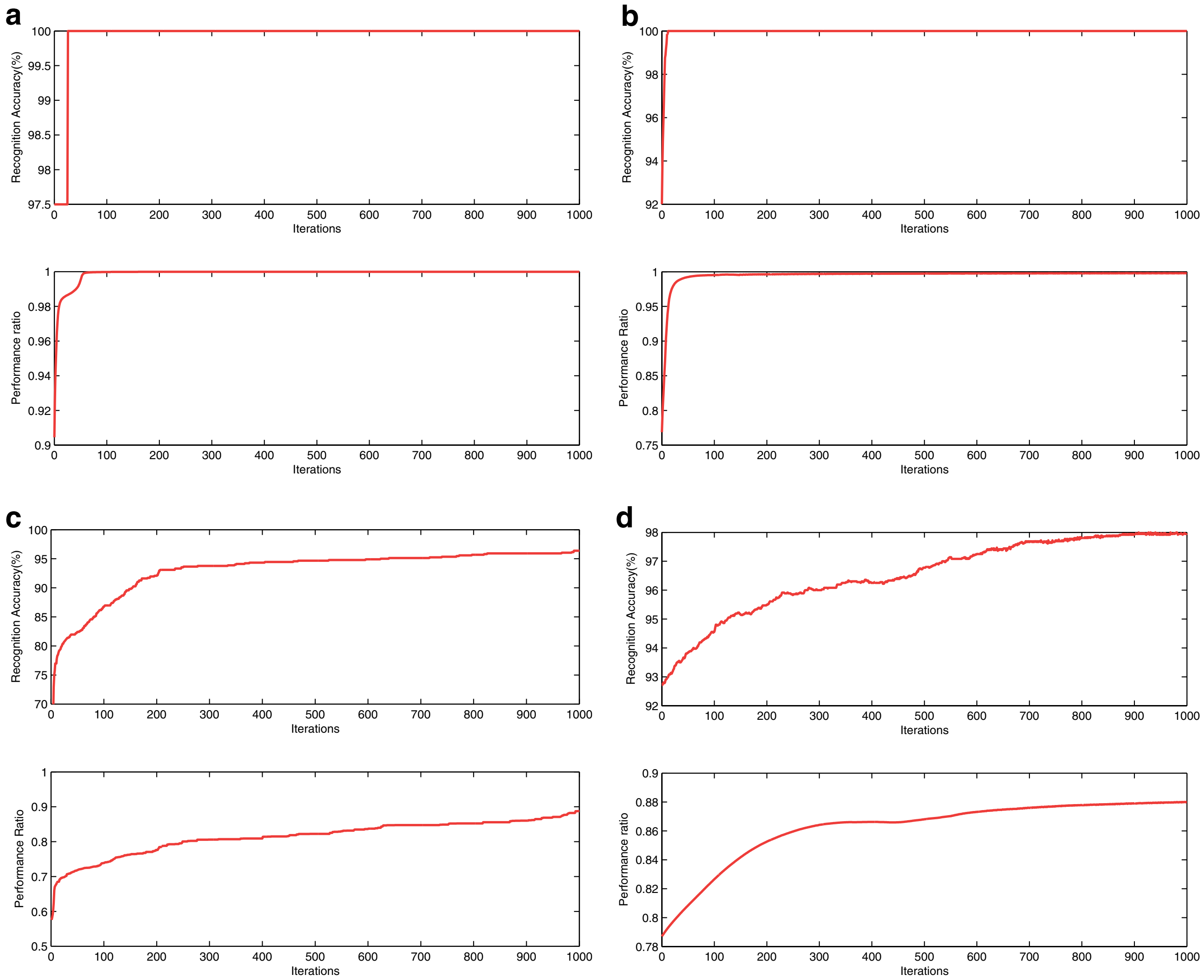

Fig. 5. Evolution of recognition accuracy and performance F versus t of PCA/OCA algorithm on the four data sets. In each panel, the top plot shows the recognition accuracy of U_t and the bottom plot shows F of U_t. (a) *ORL*; (b) *PIE*; (c) *AR*; (d) *COIL*.

performance is estimated by using 10-fold cross-validation as used in [20]. In our two-stage OCA experiments, we use the same experiment setting and the same input images used in [20]. Table 4 shows the classification accuracy results of different dimensional reduction methods on *ORL* and *AR* face set. The mean and standard deviation (in parenthesis) of accuracies from 10 runs are shown. (For the original ORL algorithm, only a single run is conducted due to the long running time.) It shows that the proposed two-stage OCA method outperforms other methods in all the cases.

5. Conclusion

In this paper we have proposed a family of two-stage OCA algorithms to improve the search efficiency and reduce the computation required by the original OCA algorithm at the same time. These algorithms are derived based on the observation that the much of solution space of the original OCA corresponds the null space of datasets and therefore the search there is not effective for typical recognition applications. By using a dimension reduction method first, the learning efficiency in the solution space is greatly improved. The significant improvement in efficiency and effectiveness is also supported by experiments using four recognition data sets. Additionally, compared to other methods, the proposed two-stage OCA algorithms often give marked improvement in classification accuracy.

Note that two-stage OCA requires n_1 (the dimension given by the first stage) to be given. It seems that an optimal range of values exists: when n_1 is too large (in the extreme case it becomes the original OCA), the complexity increases and the search becomes ineffective; on the other hand, when n_1 is too small (in the extreme case, the second stage OCA is not required and it degenerates to the first-stage method), effective linear representations may not be obtained (e.g., PCA for the *AR* dataset in Table 4). While one can estimate for the lowest dimension that gives satisfactory performance by a linear search, it seems that a better solution is to generalize the proposed algorithm to

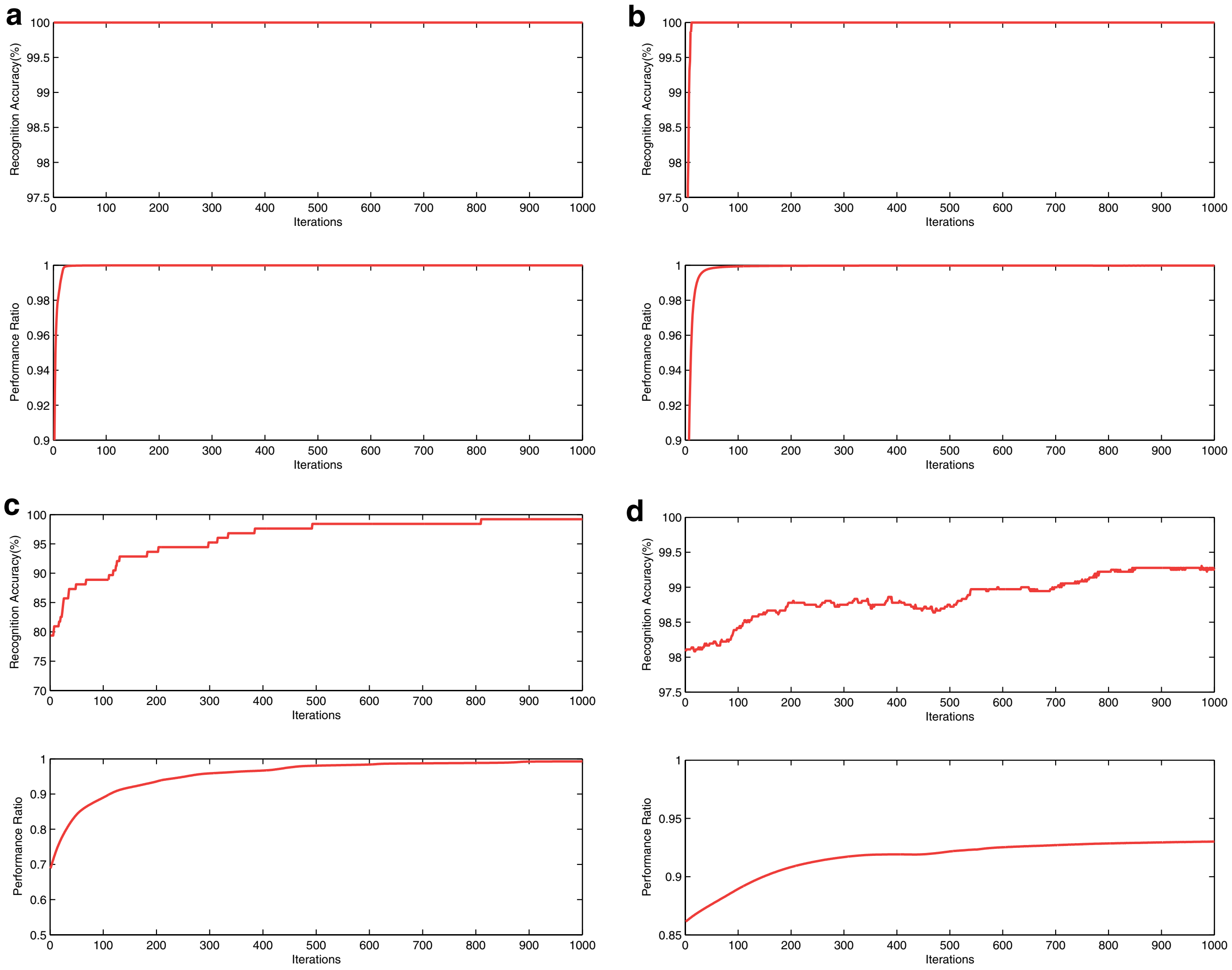

Fig. 6. Evolution of recognition accuracy and performance F versus t of LDA/OCA algorithm on the four data sets. In each panel, the top plot shows the recognition accuracy of U_t and the bottom plot shows F of U_t. (a) *ORL*; (b) *PIE*; (c) *AR*; (d) *COIL*.

multiple stages, i.e., one can learn a solution in a lower reduced space and then using the found solution to initialize the search in a higher space, which can be done recursively. The effectiveness of multiple stage OCA algorithms is being studied.

While this paper focuses on linear representations, the proposed two-stage OCA algorithms can be generalized to model nonlinearity using kernel methods. As shown in [31], nonlinear representations induced by a kernel function can be written as linear representations with respect to a basis that depends on the kernel function and the training set. As two-stage OCA algorithms significantly improve the performance of OCA, two-stage OCA algorithms with respect to the kernel basis should improve kernel OCA; this is being investigated.

Acknowledgments

The authors thank the anonymous reviewers whose comments have improved the presentation of this paper significantly and Editor K.W. Bowyer for handling this paper. The authors also thank the producers of the *ORL*, *PIE*, *AR* and *COIL* data sets for making them public. This research was supported in part by NSF Grants CCF-0514743, IIS-0307998, ACI-0324944 and ARO Grant W911NF-04-01-0268.

References

[1] T.B. Nguyen, B.J. Oommen, Moment-preserving piecewise linear approximation of signals and images, IEEE Transaction on Pattern Analysis and Machine Intelligence 19 (1997) 84–91.

[2] X. He, S. Yan, Y. Hu, P. Niyogi, H. Zhang, Face recognition using Laplacianfaces, IEEE Transaction on Pattern Analysis and Machine Intelligence 27 (2005) 328–340.

[3] I.T. Jolliffe, Principal Component Analysis, Springer-Verlag, 1986.

[4] M. Turk, A. Pentland, Eigenfaces for recognition, Journal of Cognitive Neuro-science 3 (1991) 71–86.

[5] F.D. Torre, M. Black, Robust principal component analysis for computer vision, in: Proceedings of International Conference on Computer Vision, 2001, pp. 362–369.

[6] R.O. Duda, P.E. Hart, D. Stock, Pattern Classification, Wiley, 2000.

[7] K. Fukunaga, Introduction to Statistical Pattern Classification, Academic Press, 1990.

[8] H. Kong, L. Wang, E. Teoh, J. Wang, R. Venkateswarlu, A Framework of 2D Fisher discriminant analysis application to face recognition with small number of training samples, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2005, pp. 1083–1088.

[9] A. Hyvarinen, J. Karhunen, E. Oja, Independent Component Analysis, John Wiley and Sons, 2001.

[10] S.Z. Li, X. Lu, X. Hou, X. Peng, Q. Cheng, Learning multiview face subspace and facial pose estimation using independent component analysis, IEEE Transaction on Image Processing 14 (6) (2005) 705–712.

[11] A.M. Matinez, A.C. Kak, PCA versus LDA, IEEE Transaction on Pattern Analysis and Machine Intelligence 23 (2) (2001) 228–233.

[12] K. Delac, M. Grgic, S. Grgic, Independent comparative study of PCA, ICA and LDA on the FERET data set, International Journal of Imaging Systems and Technology 15 (2006) 252–260.

[13] W.J. Krzanowski, P. Jonathan, W.V. McCarthy, M.R. Thomas, Discriminant analysis with singular covariance matrices: methods and applications to spectroscopic data, Applied Statistics 44 (1995) 101–115.

[14] J. Ye, R. Janardan, C.H. Park, H. Park, An optimization criterion for generalized discriminant analysis on undersampled problems, IEEE Transaction on Pattern Analysis and Machine Intelligence 26 (8) (2004) 982–994.

[15] S. Raudys, R.P.W. Duin, On expected classification error of the FSisher linear classifier with pseudoinverse covariance matrix, Pattern Recognition Letter 19 (5–6) (1998) 385–392.

[16] J.H. Friedman, Regularized discriminant analysis, Journal of American Statistical Association 84 (405) (1989) 165–175.

[17] P.N. Belhumeour, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, IEEE Transaction on Pattern Analysis and Machine Intelligence 19 (7) (1997) 711–720.

[18] D.L. Swets, J. Weng, Using discriminant Eigenfeatures for image retrieval, IEEE Transaction on Pattern Analysis and Machine Intelligence 18 (8) (1996) 831–836.

[19] P. Howland, M. Joen, H. Park, Structure preserving dimensional reduction for clustered text data based on the generalized singular value decomposition, SIAM J. Matrix Analysis and Applications 25 (1) (2003) 165–179.

[20] J. Ye, Q. Li, A two-stage linear discriminant analysis via QR-decomposition, IEEE Transaction on Pattern Analysis and Machine Intelligence 27 (6) (2005) 929–941.

[21] A. Srivastava, X. Liu, U. Grenander, Universal analytical forms for modeling image probabilities, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (9) (2002) 1200–1214.

[22] X. Liu, A. Srivastava, K. Gallivan, Optimal linear representation of images for object recognition, IEEE Transaction on Pattern Analysis and Machine Intelligence 26 (5) (2004) 662–666.

[23] A. Srivastava, X. Liu, Tools for application-driven dimensional reduction, Neural Computing 67 (2005) 136–160.

[24] Y. Wu, X. Liu, W. Mio, K.A. Gallivan, Two-stage optimal component analysis, in: Proceedings of IEEE International Conference on Image Processing, 2006, pp. 2041–2044.

[25] X. Liu, A. Srivastava, D.L. Wang, Intrinsic generalization analysis for low dimensional representation, Neural Networks 16 (5/6) (2003) 537–545.

[26] A. Edelman, T. Arias, S.T. Smith, The geometry of algorithms with orthogonality constraints, SIAM Journal on Matrix Analysis and Applications 20 (2) (1998) 303–353.

[27] F. Samaria, A. Harter, Parameterization of a stochastic model for human face identification, in: Proceedings of 2nd IEEE Workshop on Applications of Computer Vision, 1994.

[28] Terence Sim, Simon Baker, Maan Bsat, The CMU Pose, Illumination, and Expression (PIE) Database of Human Faces, Tech. Rep. CMU-RI-TR-01-02, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, January 2001.

[29] A.M. Martinez, R. Benavente, The AR Face Database, CVC Technical Report No. 24, 1998.

[30] S.K. Nayar, S.A. Nene, H. Murase, Columbia Object Image Library (coil-100), Technical Report CUCS-006-96, 1996.

[31] X. Liu, W. Mio, Kernel methods for nonlinear discriminative analysis, in: Proceedings of the International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition, 2005, pp. 584–599.