Incremental PCA-LDA Algorithm

Issam Dagher dagheri@balamand.edu.lb
Department Of Computer Engineering
University of Balamand
POBOX 100,Elkoura,Lebanon

Abstract

In this paper a recursive algorithm of calculating the discriminant features of the PCA-LDA procedure is introduced. This algorithm computes the principal components of a sequence of vectors incrementally without estimating the covariance matrix (so covariance-free) and at the same time computing the linear discriminant directions along which the classes are well separated. Two major techniques are used sequentially in a real time fashion in order to obtain the most efficient and linearly discriminative components. This procedure is done by merging the runs of two algorithms based on principal component analysis (PCA) and linear discriminant analysis (LDA) running sequentially. This algorithm is applied to face recognition problem. Simulation results on different databases showed high average success rate of this algorithm compared to PCA and LDA algorithms. The advantage of the incremental property of this algorithm compared to the batch PCA-LDA is also shown.

Keywords: Recursive PCA-LDA, principal component analysis (PCA), linear discriminant analysis (LDA), face recognition.

1. INTRODUCTION

A large number of face recognition techniques use face representations found by unsupervised statistical methods. Typically, these methods find a set of basis images and represent faces as a linear combination of those images. For the same purpose, this paper merges sequentially two techniques based on principal component analysis and linear discriminant analysis.
The first technique is called incremental principal component analysis (IPCA) which is an incremental version of the popular unsupervised principal component technique. The traditional PCA algorithm [13] computes eigenvectors and eigenvalues for a sample covariance matrix derived from a well known given image data matrix, by solving an eigenvalue system problem. Also, this algorithm requires that the image data matrix be available before solving the problem (batch method). The incremental principal component method updates the eigenvectors each time a new image is introduced.
The second technique is called linear discriminant analysis (LDA) [14]. LDA is a data separation technique. The objective of LDA is to find the directions that will well separate the different classes of the data once projected upon. The set of human faces is represented as a data matrix X where each row corresponds to a different human face.

Each image x, represented by a (n,m) matrix of pixels, will be represented by a high dimensional vector of nxm pixels. Turk and Pentland [22] were among the first who used this representation for face recognition.

2-dimensional principal component analysis (2dPCA) [31] was proposed which directly computes the eigenvectors of the covariance matrix without matrix to vector conversion.
2-dimensional LDA [32,33] computes directly the directions which will separate the classes without matrix to vector conversion as well.
Higher recognition rate was reported for both cases.
Both of theses algorithms work in batch mode, all the image data must be present a priori.

It should be noted that incremental face recognition systems are studied in [3,7,8,9,10,11,12], incremental LDA was studied in [1,2,4,5,6,].
While the incremental PCA methods update the eigenspace model (the covariance matrix), the incremental LDA methods update the fisherspace model (the within class scatter matrix and the between class scatter matrix). The incremental PCA-LDA updates directly the eigenvectors using the general eigenvalue problem and the complementary space. It also updates the fisherspace model based on the projected PCA data and an update of the inverse of the within class scatter matrix.

The PCA/LDA-based face recognition systems suffer from the scalability problem. To overcome this limitation, an incremental approach is a natural solution. The main difficulty in developing the incremental PCA/LDA is to compute covariance matrix and to handle the inverse of the within-class scatter matrix.
These techniques have been applied to 3D object recognition [17], sign recognition [18], and autonomous navigation [19] among many other image analysis problems [16,26,27,28]. However, the batch method no longer satisfies an up coming new trend of computer vision research [20] in which all 2-d filters are incrementally derived from very long online real-time video stream. Online development of 2-d filters requires that the system perform while new sensory signals flow in. When the dimension of the image is high, both the computation and storage complexity grow dramatically. Thus, the idea of using a real time process becomes very efficient in order to compute the principal components for observations (faces) arriving sequentially

It should be noted that the incremental PCA-LDA has the following advantages:

1. Low memory demands: No need to store all the images (mainly due to the incremental structure of the PCA). All you need to store are the eigenvectors. Given a new image or a new class, the eigenvectors will be updated using only the stored eigenvectors. From a practical point of view, there is no need to store any face database (store the unrecognized eigenvectors) and some image data could not be presently available. It should be noted that 2dPCA, 2dLDA, and SVD work in batch mode.
2. Low computational complexity: the batch PCA-LDA needs to compute all the eigenvectors of all the data then gets the first k eigenvectors. The incremental PCA-LDA operates directly on the first k eigenvectors (unwanted vectors do not need to be calculated). The processing of IPCA_LDA is restricted to only the specified number of k directions and not on all the directions.
3. Better recognition accuracy and less execution time.
4. Updating the inverse of the within class scatter matrix without calculating its inverse.

2. DERIVATION OF THE ALGORITHM

Given that large number of d-dimensional vectors, **u**(1); **u**(2); ... , which are the observations from a certain given image data, are received sequentially. The class of each image is determined. Without loss of generality, a fixed estimated mean image μ is initialized in the beginning of the algorithm. It should be noted that a simple way of getting the mean image is to present sequentially all the images and calculating their mean. At the same time, the mean μ_j and the number of data n_j of each class are determined. The mean image can be subtracted from each vector **u**(n) in order to obtain a normalization vector of approximately zero mean. Let C = $E\left[u(n)u^T(n)\right]$ be the dxd covariance matrix.

The proposed algorithm takes the number of input images (n), the dimension of the images, and the number of desired PCA (k) directions as inputs and returns the c-1 LDA vectors (c-1 directions that will well separate the different classes of the data once projected upon.) and the image coordinates with respect to these vectors as outputs. It works like a linear system that predicts the next state vector from an input vector and a current state vector. All the components will be updated from the previous values and from a new input image vector by processing sequentially the PCA and LDA algorithms. While incremental PCA returns the estimated eigenvectors as a matrix that represents subspaces of data and the corresponding eigenvalues as a row vector, incremental LDA searches for the directions where the projections of the input data vectors are well separated.

The obtained vectors will form a basis which describes the original data set without loss of information. The face recognition can be done by projecting the input test image onto this basis and comparing the resulting coordinates with those of the training images in order to find the nearest appropriate image.

A. Batch PCA-LDA algorithm :

Given [21] the image data Ui,

(Steps 1-4 compute the k PCA eigenvectors. Step 5 computes the projected LDA data on those eigenvectors. Steps 6-8 compute the LDA directions which separate the data.)

1. Subtract the sample mean from the data: $Y_i = U_i - \mu \qquad i = 1,2,....,n$
2. Compute the scatter matrix S: $S = \sum_{i=1}^{n} Y_i Y_i^T$
3. Compute eigenvectors { ***v****1*, ***v****2* …**v**k } corresponding to the largest k eigenvalues of S.
4. Let ***v****1*, ***v****2* …**v**k *be the columns of eigenvector matrix A= [**v**1, **v**2 …**v**k].*
5. The new projected LDA data are: $Z_i = A^T Y_i \qquad i = 1,2,...,n$
6. Compute the sample mean μ_Z of the LDA data and the sample mean μ_{ZJ} of each class.
7. Compute the between class scatter matrix Sb and the within class scatter matrix Sw.

$$S_b = \sum_{i=1}^{c} n_i(\mu_{zi} - \mu_z)(\mu_{zi} - \mu_z)^T; \qquad S_w = \sum_{i=1}^{c} \sum_{\text{class } k} (Z_k - \mu_{zi})(Z_k - \mu_{zi})^T$$

8. Solve $S_b w = \lambda S_w w \Rightarrow [w_1, w_2, ..., w_{c-1}]$

B. Incremental PCA equations:

By definition [10], an eigenvector **x** with a corresponding eigenvalue λ of a covariance matrix C satisfies:

$$\lambda.x = C.x \qquad (1)$$

By replacing in (2) the unknown C with the sample covariance matrix $\frac{1}{n}\sum_{i=1}^{n} u(i).u^T(i)$ and using **v** = λ.**x**, the following equation is obtained:

$$v(n) = \frac{1}{n}\sum_{i=1}^{n} u(i).u^T(i).x(i) \qquad (2)$$

where **v**(n) is the nth step estimate of **v** after entering all the n images.
Since λ = ||**v**|| and **x** = **v**/||**v**||, **x**(i) is set to **v**(i-1)/||**v**(i-1)|| (estimating **x**(i) according to the given previous value of **v**). Equation (2) leads to the following equation:

$$v(n) = \frac{1}{n}\sum_{i=1}^{n} u(i).u^T(i).\frac{v(i-1)}{\|v(i-1)\|} \qquad (3)$$

Equation (3) can be written in a recursive form:

$$v(n) = \frac{n-1}{n}v(n-1) + \frac{1}{n}u(n)u^T(n)\frac{v(n-1)}{\|v(n-1)\|} \qquad (4)$$

where $\frac{n-1}{n}$ is the weight for the last estimate and $\frac{1}{n}$ is the weight for the new data.
To begin with, Let's set **v**(0) = **u**(1), the first direction of data spread. The IPCA algorithm will give the first estimate of the first principal component $v(1)$ that corresponds to the maximum eigenvalue:

$$v(1) = \frac{1}{n}u(1)u^T(1)\frac{v(0)}{\|v(0)\|}$$

Each time a new image is introduced, the eigenvectors will be updated. They are presented by the algorithm in a decreasing order with respect to the corresponding eigenvalue (the first eigenvector will correspond to the largest eigenvalue). The other higher order vectors are estimated following what *Stochastic Gradient Ascent* SGA does: Start with a set of orthonormalized vectors, update them using the suggested iteration step, and recover the orthogonality using *Gram-Schmidt Orthonormalization* GSO. For real-time online computation, avoiding time-consuming GSO is needed. Further, the vectors should be orthogonal to each other in order to ensure the independency. So, it helps to generate “observations” only in a complementary space for the computation of the higher order eigenvectors. For example, to compute the second order vector, first the data is subtracted from its projection on the estimated first order eigenvector $v_1(n)$, as shown in (6):

$$u_2(n) = u_1(n) - u_1^T(n)\frac{v_1(n)}{\|v_1(n)\|}\frac{v_2(n)}{\|v_2(n)\|} \qquad (5)$$

where $\mathbf{u}_1(n)$ = **u**(n). The obtained residual, $\mathbf{u}_2(n)$, which is in the complementary space of $\mathbf{v}_1(n)$, serves as the input data to the iteration step.

In this way, the orthogonality is always enforced when the convergence is reached, although not exactly so at early stages. This, in effect, better uses the sample available and avoids the time-consuming GSO.
After convergence, the vectors will also be enforced to be orthogonal, since they are estimated in complementary spaces.

C. Incremental LDA equations :

At iterartion k (corresponding to the data U_k):

Given the projected LDA data $Z_k = AY_k$

and the sample mean of class j $\mu_{zj} = \frac{\sum_{class\ j} Z_k}{n_j} = A\frac{\sum_{class\ j} Y_k}{n_j} = A\frac{\sum_{class\ j}(X_k - \mu)}{n_j} = A(\mu_j - \mu)$

and the general following formula : $(D+BC)^{-1} = D^{-1} - D^{-1}B(I + CD^{-1}B)^{-1}CD^{-1}$

The update formulas for S_w is given by:

$$S_w = S_w + (Z_k - \mu_{kj})(Z_k - \mu_{kj})^T = S_w + A(Y_k - (\mu_j - \mu))(Y_k - (\mu_j - \mu))^T A^T$$

Using the previous 2 equations the update formula for the inverse of Sw is:

$$S_{wi} = S_{wi} - \frac{S_{wi}A(Y_k - (\mu_j - \mu))(Y_k - (\mu_j - \mu))^T A^T S_{wi}}{1 + (Y_k - (\mu_j - \mu))^T A^T S_{wi}(Y_k - (\mu_j - \mu))A}$$

Because A is not yet completely specified, the following update matrices will be used:

$$S_{w1} = S_{w1} + (Y_k - (\mu_j - \mu))(Y_k - (\mu_j - \mu))^T$$

$$S_{wi1} = S_{wi1} - \frac{S_{wi1}(Y_k - (\mu_j - \mu))(Y_k - (\mu_j - \mu))^T S_{wi1}}{1 + (Y_k - (\mu_j - \mu))^T S_{wi1}(Y_k - (\mu_j - \mu))}$$

And at the end of the nth iteration:

$$S_w = AS_{w1}A^T$$

$$S_{wi} = AS_{wi1}A^T$$

$$S_b = \sum_{i=1}^{c} n_i(\mu_{zi} - \mu_z)(\mu_{zi} - \mu_z)^T =$$

$$[w_1, w_2, ..., w_{c-1}] = \textit{eigenvectors}\ \text{of}\ (S_{wi}S_b)$$

It should be noted that the above equations are considered to be a contribution of this paper

D. Algorithm Summary:

Assume n different images **u**(n) are given.

μ is the sample mean of all the images, μ_j is the sample mean of each class,

and n_j is the number of data of each class.

Combining IPCA and LDA algorithms, the new algorithm can be summarized as follows:
(Input one image at a time. Update the k eigenvectors (v) according to that image. Update the fisherspace model based on the projected data and update the inverse of the within class scatter matrix. Compute the LDA directions (w))

Initialize Sw to zero elements and Swi to random big numbers.
For i=1:n

img = input image from image data matrix;

u(i) = img;

for j=1:k

if j == i, initialize the eigenvector as:

$$v_j(i) = u(i);$$

else

$$v_j(i) = \frac{i-1}{i} v_j(i-1) + \frac{1}{i} u(i) u^T(i) \frac{v_j(i-1)}{\|v_j(i-1)\|}; \quad \text{(vector update)}$$

$$u(i) = u(i) - u^T(i). \frac{v_j(i)}{\|v_j(i)\|} \frac{v_j(i)}{\|v_j(i)\|}; \quad \text{(To ensure orthogonality)}$$

end

end

$$Y_i = img - \mu;$$

$$S_{w1} = S_{w1} + (Y_k - (\mu_j - \mu))(Y_k - (\mu_j - \mu))^T$$

$$S_{wi1} = S_{wi1} - \frac{S_{wi1}(Y_k - (\mu_j - \mu))(Y_k - (\mu_j - \mu))^T S_{wi1}}{1 + (Y_k - (\mu_j - \mu))^T S_{w1i}(Y_k - (\mu_j - \mu))}$$

end

$$S_w = A S_{w1} A^T$$

$$S_{wi} = A S_{wi1} A^T$$

$$S_b = A\left(\sum_{i=1}^{c} n_i (\mu_i - \mu)(\mu_i - \mu)^T\right) A^T$$

$$[w_1, w_2, ..., w_{c-1}] = eigenvectors \text{ of } (S_{wi} S_b)$$

E . Comparison with PCA-LDA Batch algorithm:

The major difference between this algorithm and the PCA-LDA batch algorithm [21],[29], is the sequential flow of input data. Incremental PCA-LDA doesn't need a large memory to store the whole data matrix that represents the incoming images. Thus in each step, this function deals with one incoming image in order to update the estimated directions, and the next incoming image can be stored over the previous one. The first estimated vectors (corresponding to the largest eigenvalues) in incremental PCA correspond to the vectors that carry the most efficient information. As a result, the processing of incremental PCA-LDA can be restricted to only a specified number of first eigenvectors. On the other side, the decision of efficient vectors in PCA can be done only after calculating all the vectors, so the program will spend a certain time calculating unwanted vectors. Also, LDA works usually in a batch mode where the extraction of discriminative components of the input eigenvectors can be done only when these eigenvectors are present simultaneously at the input. It is very clear that from the time efficiency concern, incremental PCA-LDA will be more efficient and requires less execution time than batch PCA-LDA algorithm. It should also be noted that this algorithm has the advantage of updating the inverse of the within class scatter matrix without calculating its inverse.

3. Experimental Results and Discussions

A. Face recognition Evaluated by Nearest Neighbor Algorithm:

The nearest neighbor algorithm was used to evaluate the face recognition technique. The following L1 similarity measure (inner product formula) was adopted:

$$L1 = \sum_i | w_{test}(i) - w_{train}(i) \qquad (6)$$

It should be noted that the incremental PCA-LDA algorithm will give small size basis, with respect to the number of training input images. Other similarity measures could have been used and similar comparative results will be obtained.

Each Face Database was split into two sets. The training set that contains images used to calculate the discriminative vectors and come up with the appropriate basis. And the test set that contains images to be tested by the face recognition algorithm in order to evaluate the performance of the proposed method. The whole set of training images (rows in the image data matrix) is projected into the basis found in order to calculate the coordinates of each image with respect to the basis v_{train} . Each new testing image v_{test} is compared to whole set of training images v_{train} in order to come up with nearest one that corresponds to the minimum L1 in equation (6). The generalization performance (or % accuracy) is equal to the number of correctly classified testing images divided by the total number of testing images (multiplied by 100).

B. Face Recognition Performance:

Three popular face databases were used to demonstrate the effectiveness of the proposed PCA-LDA algorithm.

The ORL [23] contains a set of faces taken between April 1992 and April 1994 at the Olivetti Research Laboratory in Cambridge. It contains 40 distinct persons with 10 images per person. The images are taken at different time instances, with varying lighting conditions, facial expressions and facial details (glasses/no-glasses). All persons are in the up-right, frontal position, with tolerance for some side movement.

The UMIST [24] taken from the University of Manchester Institute of Science and Technology. It is a multi-view database, consisting of 575 images of 20 people, each covering a wide range of poses from profile to frontal views.

The Yale [25] taken from the Yale Center for Computational Vision and Control. It consists of images from 15 different people, using 11 images from each person, for a total of 165 images. The images contain variations with following total expressions or configurations: center-light, with glasses, happy, left-light, without glasses, normal, right-light, sad, sleepy, surprised, and wink.

And the BIOID database [33]. The dataset consists of 1521 gray level images with a resolution of 384x286 pixel. Each one shows the frontal view of a face of one out of 23 different test persons.

Each image in the ORL database is scaled into (92 × 112), in the UMIST Database is scaled into (112 × 92), the Yale Database is cropped and scaled into (126 × 152) and the BIOID is cropped and scaled to (128 x 95)

Figures 1,2, 3, and 4 show a sample of 6 images from all the 4 databases.

Fig. 1. Some sample images of 6 persons in the ORL Database

Fig. 2. Some sample images of 6 persons in the UMIST Database

Fig. 3. Some sample images of 6 persons in the YALE Database

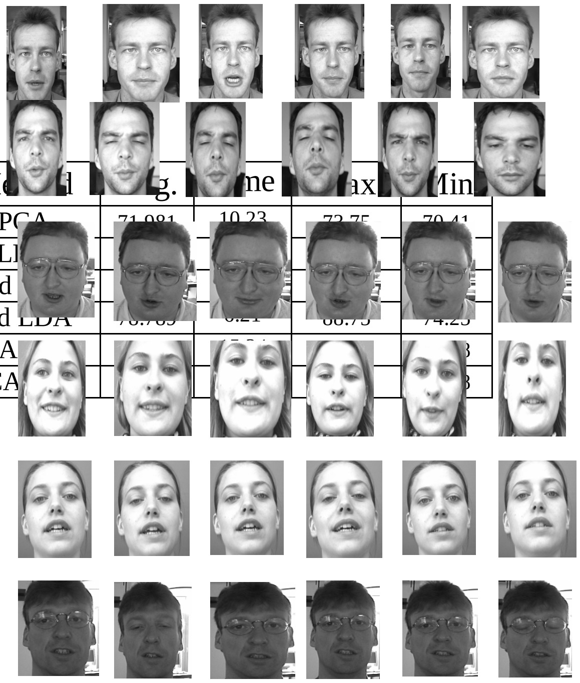

Fig. 4. Some sample images of 6 persons in the BioidDatabase

To start the face recognition experiments, each one of the three databases is randomly partitioned into a training set and a test set with no overlap between the two. 10 different partitions were made. A comparison between the average performances, Matlab time in seconds on our machine Pentium(R) D CPU 3.4 Ghz, the minimum and maximum generalization performances of these algorithms using the four databases is shown in Table 1 for 40%, 60% training and 10-fold cross validation. Figure5 shows examples from all the databases of both successful (the first 4 faces) and failed recognized images.

Fig. 5: Examples of both successful and failed recognized images.

Table1 shows for all the databases, the average (over 10 runs), the minimum, the maximum generalization performance, and the time (in sec) for 40%,60%, and 10-fold cross validation of the training database.

2-d PCA and 2-d LDA can be considered as 2-d filters which operate directly on the 2-d image.

Table1: Comparisons between time and generalization performances for the 4 databases

ORL	Method	Avg.	Time	Max	Min
40%	PCA	71.981	10.23	73.75	70.41
	LDA	[illegible]	[illegible]	[illegible]	[illegible]
	2-d PCA	[illegible]	[illegible]	[illegible]	[illegible]
	2-d LDA	78.789	8.21	88.75	74.25
	PCA-LDA	[illegible]	[illegible]	[illegible]	[illegible]
	IPCA-LDA	[illegible]	[illegible]	[illegible]	[illegible]
60%	PCA	72.25	11.77	77.51	70
	LDA	86.808	11.36	88.12	83.73
	2-d PCA	89.9	8.88	92.65	71.45
	2-d LDA	91.72	7.98	95.62	76.87
	PCA-LDA	92.187	14.92	95.5	90
	IPCA-LDA	92.062	5.57	95.5	90
10-fold	PCA	82.823	13.77	86.98	80

CV	LDA	86	13.67	88.75	70
	2-d PCA	89.67	10.92	92.35	82.23
	2-d LDA	92.45	9.11	95.65	85.65
	PCA-LDA	95	18.80	95.54	91.25
	IPCA-LDA	95.825	7.92	96.5	93.75

UMIST	Method	Avg.	Time	Max	Min
40%	PCA	53.088	12.34	56.25	50.83
	LDA	53.622	12.44	57.5	50
	2-d PCA	61.25	8.65	68.87	52.25
	2-d LDA	63.77	6.21	69.22	51.23
	PCA-LDA	63.655	16.78	69.5	60.33
	IPCA-LDA	64.155	5.57	69.5	60.41
60%	PCA	74.935	13.04	77.5	71.87
	LDA	56.185	13.01	63.12	52.5
	2-d PCA	76.94	8.33	78.87	72.54
	2-d LDA	64.31	6.57	66.56	56.70
	PCA-LDA	64.31	17.89	66.87	62.5
	IPCA-LDA	64.81	6.01	67.5	62.5
10-fold CV	PCA	75.25	14.89	80	71.25
	LDA	73.5	14.73	77.5	70
	2-d PCA	80.67	11.56	82.23	73.34
	2-d LDA	82.57	10.87	87.25	71.23
	PCA-LDA	83.5	19.11	87.5	80
	IPCA-LDA	83.5	9.11	87.5	80

YALE	Method	Avg.	Time	Max	Min
40%	PCA	64.281	6.23	79.52	45.71
	LDA	60.614	6.27	69.52	45.23
	2-d PCA	66.76	4.73	80.52	47.65
	2-d LDA	68.87	4.21	82.78	53.37
	PCA-LDA	70.595	8.34	81.52	55.23
	IPCALDA	71.35	4.11	80.52	56.21
60%	PCA	65.934	7.01	79.33	54
	LDA	60.932	7.07	73.33	50
	2-d PCA	70.76	4.98	82.76	62.33
	2-d LDA	74.21	4.67	84.25	55.21
	PCA-LDA	74.236	8.89	85.71	61.33
	IPCALDA	74.236	4.23	85.71	61.33
10-fold CV	PCA	67.778	8.11	78.89	35.56
	LDA	65.891	8.09	77.78	30
	2-d PCA	68.67	5.98	82.23	35.56
	2-d LDA	69.23	5.01	80.23	30
	PCA-LDA	70.67	9.23	85.78	30
	IPCALDA	72.576	4.98	87.78	30

BIOID	Method	Avg.	Time	Max	Min
40%	PCA	60.671	18.67	66.59	54.27
	LDA	63.178	18.54	67.89	60.11
	2-d PCA	71.576	12.33	78.11	62.37
	2-d LDA	73.89	9.21	79.78	61.65
	PCA-LDA	74.123	26.80	79.25	68.55
	IPCA-LDA	75.01	8.57	79.5	69.64
60%	PCA	62.34	19.04	66.11	59.12
	LDA	64.879	19.01	73.12	60.76
	2-d PCA	67.11	13.56	77.28	62.33
	2-d LDA	70.11	10.11	78.98	67.12
	PCA-LDA	71.12	27.96	76.13	67.34
	IPCA-LDA	72.45	9.23	77.67	68.5
10-fold CV	PCA	74.78	20.76	81.23	70.87
	LDA	76.11	20.64	82.88	72.31
	2-d PCA	80.67	14.21	83.76	70.45
	2-d LDA	82.57	11.66	88.66	80.78
	PCA-LDA	85.5	29.22	89.88	82.7
	IPCA-LDA	86.67	11.01	90.12	82.7

It should be noted that the incremental PCA-LDA has slightly higher performance than the batch PCA-LDA. This is due to the fact that the first estimated vectors (corresponding to the largest eigenvalues) in incremental PCA-LDA correspond to the vectors that carry the “most” efficient discriminant vectors. The incremental PCA-LDA concentrates first from the whole data on getting the first eigenvector and then projects to it the data in order to get the LDA vectors. So there is a correlation between the first eigenvector (independent of the other eigenvectors) and the LDA data.This is true for the second, the third, till the kth eigenvector. On the other hand the batch PCA-LDA gets first all the eigenvectors and then projects the data.
For our experiments, the number of PCA eigenvectors used for both algorithms are the number of different classes (c) and eventually the number of the LDA features are c-1 (which is the maximum number of possible features). Our experiments show that as you increase the number of features, the recognition performance increases as well. For example for the Yale database there are 15 different persons (c=15), for the UMIST database there are 20 classes (c=20).
The performance of the incremental PCA-LDA was also assessed using 2 types of error: The false acceptance rate (FAR) and false rejection rate (FRR). For all the databases 4 different persons are considered as imposters and the training is done using 80 % of the other persons. It should be noted that these 2 measures are ratios (no unit of measure is given).The following results were obtained:
It should be noted that Table1 and Table2 compares incremental PCA-LDA to 4 related algorithms: PCA, LDA, 2-d PCA, 2-d LDA, and batch PCA_LDA. This comparison is done in Table1 using the recognition performance (Average, minimum, and maximum) for different training sizes and the time in seconds of each algorithm. Table 2 assesses the performance of these algorithms using the FAR and the FRR measures.

Table2: FAR and FRR for the 4 databases.

ORL	**FAR**	**FRR**
PCA	0.103	0.0388
LDA	0.0811	0.0166
2-d PCA	0.065	0.0145
2-d LDA	0.045	0.0134
Batch PCA-LDA	0.029	0.0130
I PCA-LDA	0.027	0.0122

UMIST	**FAR**	**FRR**
PCA	0.381	0.122
LDA	0.211	0.201
2-d PCA	0.178	0.145
2-d LDA	0.158	0.107
Batch PCA-LDA	0.156	0.106
IPCA-LDA	0.123	0.102

YALE	FAR	FRR
PCA	0.561	0.478
LDA	0.478	0.321
2-d PCA	0.356	0.287
2-d LDA	0.301	0.245
Batch PCA-LDA	0.311	0.234
I PCA-LDA	0.247	0.189

BIOID	FAR	FRR
PCA	0.336	0.178
LDA	0.234	0.189
2-d PCA	0.167	0.11
2-d LDA	0.132	0.107
Batch PCA-LDA	0.133	0.108
I PCA-LDA	0.124	0.105

As stated before, LDA has the supervised property over PCA. It uses the class labels in order to get the optimum directions which separates the data. As the tables show the performances of LDA is not always better than PCA. This is due mainly to the size of the training set. In the non representative (small) training set PCA outperforms LDA. In that case, the PCA vectors are better than the LDA vectors. To overcome this problem is to use the two-stage feature extraction method PCA followed by LDA. The tables show that Batch PCA-LDA and incremental PCA_LDA outperforms the PCA and LDA algorithms in %generalization (better generalization performances) and in the FAR and the FRR (less number of false acceptances and less number of false rejections).It should be noted that other mixed techniques exist in the literature. For example in [1] incremental PCA – ICA was used to transform the principle components to independent directions irrespective of the class label (unsupervised property vs. the supervised property of the LDA). Another example the PCA-support vectors uses kernel PCA and solving a quadratic optimization problem (batch mode vs. the incremental mode of this paper).

4. CONCLUSIONS

In this paper a recursive algorithm of calculating the discriminant features of the PCA-LDA procedure is introduced. The method concentrates on a challenging issue of computing discriminating vectors from an incrementally arriving high-dimensional data stream without computing the corresponding covariance matrix and without knowing the data in advance. Because real-time face identification is necessary in

most practical applications, this proposed method can process face images (including training and identifying) in high speed and obtain good results. Its effectiveness and good performance has been proven by experiments.
The proposed incremental PCA-LDA algorithm is very efficient in memory usage (only one input image is needed at every step). And it is very efficient in the calculation of the first basis vectors (unwanted vectors do not need to be calculated). In addition to these advantages, this algorithm gives an acceptable face recognition success rate in comparison with very famous face recognition algorithms such as the PCA and LDA. Even though this algorithm gave similar results to the batch PCA-LDA, it has the advantage of its simple updates of all the parameters when introducing new image in the training set. Batch PCA-LDA need to use all the training images to recalculate the new basis.

References

[1] H Zhao, PC Yuen, “Incremental Linear Discriminant Analysis for Face Recognition”, Systems, Man, and Cybernetics, Part B, IEEE Transactions on, Vol. 38, No. 1. (2008), pp. 210-221.
[2] Shaoning Pang, Seiichi Ozawa, Nikola Kasabov, “Chunk Incremental LDA Computing on Data Streams”, Lecture Notes in Computer Science (Advances in Neural Networks – ISNN 2005), Volume 3497/2005, pp. 51-56
[3] J. Ye, H. Xiong Q. Li, H. Park, R. Janardan, and V. Kumar.” IDR/QR: An Incremental Dimension Reduction Algorithm via QR Decomposition”, In Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, pages 364–373, 2004.
[4] M. Uray, D. Skočaj, P. Roth, H. Bischof and A. Leonardis, “Incremental LDA learning by combining reconstructive and discriminative approaches”, Proceedings of British Machine Vision Conference (BMVC) 2007, pp. 272-281
[5] Tae-Kyun Kim; Shu-Fai Wong; Stenger, B.; Kittler, J.; Cipolla, R., “Incremental Linear Discriminant Analysis Using Sufficient Spanning Set Approximations” Computer Vision and Pattern Recognition, 2007. IEEE Conference on, Volume , Issue , 17-22 June 2007 pp. 1-8
[6] Haitao Zhao; Pong Chi Yuen; Kwok, J.T.; Jingyu Yang,” Incremental PCA based face Recognition”, ICARCV 2004 Dec. 2004 Page(s): 687 - 691 Vol. 1
[7] Ghassabeh, Y.A.; Ghavami, A.; Moghaddam “A New Incremental Face Recognition System” Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2007. IDAACS 2007. 6-8 Sept. 2007 Page(s):335 – 340
[8] Fengxi Song, David Zhang, Qinglong Chen1, and Jingyu Yang,” A Novel Supervised Dimensionality Reduction Algorithm for Online Image Recognition” , Lecture Notes in Computer Science ; PSIVT 2006, LNCS 4319, pp. 198 – 207, 2006. Springer-Verlag
[9] Fengxi Song , Hang Liu, David Zhang, Jingyu Yang “A highly scalable incremental facial feature extraction method”, Elsevier. Neurocomputing 71 (2008) 1883– 1888
[10] Dagher, I.; Nachar, R. “Face recognition using IPCA-ICA algorithm”, Pattern Analysis and Machine Intelligence, IEEE Transactions on Volume 28, Issue 6, June 2006 Page(s):996 – 1000
[11] B. Raducanu, J. Vitria “Learning to learn: From smart machines to intelligent machines”, Pattern Recognition Letters 29 (2008) 1024–1032
[12] A Haitao Zhao Pong Chi Yuen Kwok, J.T. “novel incremental principal component analysis and its application for face recognition Systems”, Man, and Cybernetics, Part B, IEEE Transactions on. Aug. 2006, Volume: 36, Issue: 4 On page(s): 873-886
[13] J. Karhunen and J. Joutsensalo,” Representation and separation of signals using non linear PCA type learning”, Neural Networks, 7(1),1994.
[14] K. Fukunaga, “Introduction to statistical pattern recognition”, Second ed., Academic Press, 1990
[15] J. J. Atick and A. N. Redlich, “What does the retina know about natural scenes?”, *Neural Comput.*, vol. 4, pp. 196-210, 1992.
[16] 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*. August 1996
[17] H. Murase and S.K. Nayar, “Visual Learning and Recognition of 3-D Objects from

Appearance," *Int'l J. Computer Vision*, vol. 14, no. 1, pp. 5-24, Jan. 1995.
[18] Y. Cui and J. Weng, "Appearance-Base Hand Sign Recognition from Intensity Image Sequences," *Computer Vision and Image Understanding*, vol. 78, pp. 157-176, 2000.
[19] S. Chen and J. Weng, "State-Based SHOSLIF for Indoor Visual Navigation," *IEEE Trans. Neural Networks*, vol. 11, no. 6, pp. 1300-1314, 2000.
[20] J. Weng and I. Stockman, eds., *Proc. NSF/DARPA Workshop Development and Learning*, Apr. 2000.
[21] A. M. Martinez and A. C. Kak, **"***PCA versus LDA"*, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 228–233, Feb 2001
[22] M. Turk and A. Pentland, "Eigenfaces for Recognition," *J. Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.
[23] ORL face database, website: http://www.cam-orl.co.uk/facedatabase.html, *AT&T Laboratories Cambridge*.
[24] UMIST face database, website: http://images.ee.umist.ac.uk/danny/database.html, *Daniel Graham.*
[25]Yale face database,website:http://www1.cs.columbia.edu/~belhumeur/pub/images/yalefaces/, *Colubmbia University.*
[26] Bioid face database, website: *www.**bioid**.com/downloads/facedb/*
[27] *Yunhong Wang, Tieniu Tan and Yong Zhu*, "Face Verification Based on Singular Value Decomposition and Radial Basis Function Neural Network+," *Institute of Automation, Chinese Academy of Sciences, Beijing, P. R. China, 100080*.
[28] Hakan Cevikalp, Marian Neamtu, Mitch Wilkes, and Atalay Barkana, "Discriminative Common Vectors for Face Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, pp.6-9, 2005.
[29] H. Yu and J. Yang**,** "*A direct LDA algorithm for high-dimensional data—With application to face Recognition" Pattern Recognition*, vol. 34, pp. 2067–2070, 2001.
[30] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs Fisherfaces: recognition using class specific linear projection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[31] Yang J, Zhang D, Frangi A.F., and Yang J.Y. " two dimensional PCA: a new approach to appearance-based face representation and recognition" IEEE PAMI, vol 26, no 1 pp:131-137, Jan 2004.
[32] Ye J., Janardan R., and Li Q.,"two dimensional linear discriminant analysis", NIPS 2004.
[33] Xiong H., Swamy M.N.S, and Ahmad M.O." two dimensional FLD for face recognition" Pattern Recognition, vol 38, pp1121-1124, 2005.