Face Recognition Using Incremental Principal Component Analysis

Satish S. Banait[1], Vivek Kshirsagar[2], Meghana Nagori[3], Archana R. Ugale[4]
1 Dept. of Computer Engg.KK Wagh Institute of Engg. Education & Research Centre, Nashik
2, 3 Dept. of Computer Science & Engineering, Govt. College Of Engineering, Aurangabad, India
4 Dept. of Computer Engg.MET's BKC College of Engg., Nashik
banaitss@yahoo.com,vkshirsagar@gmail.com,kshirsagarmeghana@gmail.com,ar.ugale@gmail.com

ABSTRACT **- IN this paper, a new approach to face recognition is presented in which not only a classifier but also a feature space is learned incrementally to adapt to a chunk of training samples. Human face recognition plays a significant role in security applications for access control and real time video surveillance systems, and robotics. Popular approaches for face recognition, such as principal components analysis (PCA), rely on static datasets where training is carried in a batch-mode on a pre-available image set. Real world applications require that the training set be dynamic of evolving nature where within the framework of continuous learning new training images are continuously added to the original set; this would trigger a costly frequent re-computation of the eigen space representation via repeating an entire batch-based training that includes the new images. Incremental PCA methods allow adding new images and updating the PCA representation, and offer the advantage of dispensing with the recently added images after model update. A benefit of this type of incremental learning is that the search for useful features and the learning of an optimal decision boundary are carried out in an online fashion. To implement this idea, chunk incremental principal component analysis (IPCA) and resource allocating network with long-term memory are effectively combined. In this paper, various incremental PCA (IPCA) training and relearning strategies are proposed and applied to the candid covariance-free incremental principal component algorithm. The effect of the number of increments and size of the eigen vectors on the correct rate of recognition is studied.**

Keywords—IPCA-ICA, Principal component analysis (PCA), independent component analysis (ICA), principal non-Gaussian directions, image processing, blind source separation.

I. INTRODUCTION

A large number of face recognition techniques use face representations found by unsupervised statistical methods. The problem of automatic human face recognition can be stated as follows: given an image of a human face, compare it with pre-stored models of a set of face images labeled with the person's identity (the training set) and report the matching result. Face recognition is one-to-many process that compares an input test image against all face templates used in training; the output is the identity of the input test image. Typically, thesemethods find a set of basis images and represent faces as a linear combination of those images. For the same purpose, this papermerges sequentially two techniques based on principal component analysis and independent component 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[1] 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 independent component analysis (ICA) [2]. It is used to estimate the independent characterization of human faces.Atick and Redlich have argued for such representationsas a general coding strategy for the visual system [3]. It is known that there is a correlation or dependency between different human faces. Finding the independent basic faces form those correlated ones is a very important task. The set of human faces is represented as a data matrixXwhere each row corresponds to a differenthumanface. The correlation between rows of matrix X can be represented as the rows of a mixing matrix A. The independent basic faces are represented as rows of source matrix S. The ICA algorithm extracts these independent faces from a set of dependent ones using [1]. It should be noted that ICA is much related to the method called blind source separation (BSS) [4], where a correlated source is separated into uncorrelated source without previous knowledge about the correlation between the elements of the source. These techniques have been applied to 3D object recognition [5], sign recognition [6], and autonomous navigation [7] among many other image analysis problems. When the dimension of the image is high, both thecomputation and storage complexity grow dramatically. Thus, theidea of using a real-time process becomes very efficient in order tocompute the principal independent components for observations(faces) arriving sequentially. Each

eigenvector or principal componentwill be updated, using FastICA algorithm, to a non-Gaussiancomponent. It should be noted that a random vector is said to benon-Gaussian if its distribution is not a Gaussian distribution. In [1],if the source matrix S contains Gaussian uncorrelated elements, thenthe resulting elements in the mixed matrix X will be also Gaussianbut correlated elements.

$$X = A.S \quad (1)$$

The FastICA method does not have a solution if the randomvariables to estimate are Gaussian random variables. This is dueto the fact that the joint distribution of the elements of X will becompletely symmetric and doesn't give any special informationabout the columns of A. In this paper, S is always a non-Gaussian vector.

Each image x, represented by a (n,m) matrix of pixels, will berepresented by a high-dimensional vector of n × m pixels. It shouldbe noted that image intensities have non-Gaussian distribution.Simoncelli [9] found that the histograms have much heavier tailsand more sharply peaked. Turk and Pentland [10] were among thefirst who used this representation for face recognition.These image vectors will be the rows of X and the resulting uncorrelated components will be the rows of S.Therefore, each column of A,called w, will be a direction that maximizes the non-Gaussianity ofthe projection of the dependent images x into w.

II. RELATED WORK

Face recognition approaches may be categorized under two general approaches: appearance-based (holistic) and feature-based (structural). Both approaches are designed to use previous knowledge obtained from feature extraction to recognize human faces [1, 2, 3, 4]. The most popular appearance-based holistic approaches includes: (1) the eigenfaces, known also as the Principal Components Analysis (PCA) and also as Kahunen-Loeve transformation (KL) [5, 6, 7], (2) the Fisherfaces known as the linear Discriminant analysis (LDA) [8] , and (3) Independent Component Analysis (ICA) [9]. PCA is unsupervised technique for dimensionality reduction; it searches for directions in the dataset that have the largest variance and define a projection matrix to project the data onto it. This leads to a lower dimensional presentation of the data, and therefore removes some of the noisy directions. Batch mode determination of principal axes for data with varying reliability and missing data was studied in [10, 11, 12, 13].

III. DERIVATION OF THE ALGORITHM

Each time a new image is introduced, the non-Gaussian vectors will be updated. They are presented by the algorithm in a decreasing order with respect to the corresponding eigenvalue (the first non-Gaussian vector will correspond to the largest eigenvalue). While the convergence of the first non-Gaussian vector will be shown in Section 3.1, the convergence of the other vectors will be shown in Section 3.2.

A. The First Non-Gaussian Vector

1. Algorithm Definition

Suppose that the sample d-dimensional vectors, u(1); u(2); . . . ; possibly infinite, which are the observations from a certain givenimage data, are received sequentially. Without loss of generality, a fixed estimated mean image is initialized in the beginning of thealgorithm. It should be noted that a simple way of getting the mean image is to present sequentially all the images and calculating theirmean. This mean can be subtracted from each vector u(n) in order to obtain a normalization vector of approximately zero mean. Let C = $E[u(n)u^{T}(n)]$be the d×d covariance matrix, which is not known asan intermediate result. The IPCA_ICA algorithm can be described asfollows.The proposed algorithm takes the number of input images, thedimension of the images, and the number of desired non-Gaussiandirections as inputs and returns the image data matrix and the non-Gaussian vectors as outputs. It works like a linear system thatpredicts the next state vector from an input vector and a current statevector. The non-Gaussian components will be updated from theprevious components values and from a new input image vector byprocessing sequentially the IPCA and the FastICA algorithms.While IPCA returns the estimated eigenvectors as a matrix thatrepresents subspaces of data and the corresponding eigenvalues as arow vector, FastICA searches for the independent directions w,where the projections of the input data vectors will maximize thenon-Gaussianity.It is based on minimizing the approximate negentropy function given by the equation $J(x) = \sum_i k_i \{ E\left(\check{G}_i(x)\right) - \check{E}(G_i(v))\}^2$ using Newton's method.

The obtained independent vectors will form a basis which describes the original data set without loss of information.

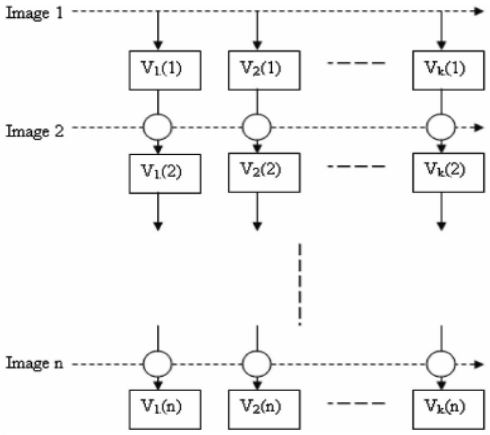


Fig.1 IPCA_PCA algorithm description

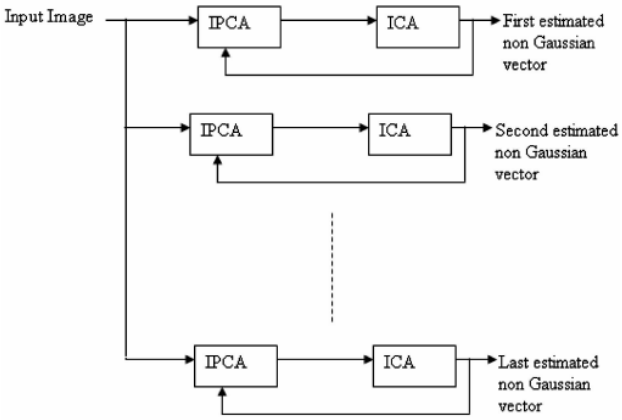


Fig. 2.IPCA ICA algorithm Block Diagram.

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. Assume the data consists of n images and a set of k non-Gaussian vectors are given, Fig. 1 illustrates the steps of the algorithm. Initially, all the non-Gaussian vectors are chosen to describe anorthonormal basis. In each step, all those vectors will be updated using an IPCA updating rule presented in (7). Then, each estimatednon-Gaussian vector will be an input for the ICA function in order to extract the corresponding non-Gaussian vector from it (Fig. 2).

2. Algorithm Equations

By definition, an eigenvector x with a corresponding eigenvalue λ of a covariance matrix C satisfies:
$\lambda . x = c . x$(2)

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 $v = \lambda . x$

The following equation is obtained:

$$u(n) = \frac{1}{n}\sum_{i=1}^{n} u(i).u^T(i)x(i) \quad (3)$$

Where$v(n)$is the n^{th} step estimate of v after entering all the n images
Since
$\lambda = \| V \| \; and X = V/\| V \|, X(i) is set to V(i-1) / \| V(i-1) \|$
(estimating $X(i)$according to the given previous value of v).

Equation (3) leads to the following equation:

$$u(n) = \frac{1}{n}\sum_{i=1}^{n} u(i).u^T(i).\frac{u(i-1)}{\|u(i-1)\|} \quad (4)$$

Equation (4) can be written in a recursive form:

$$v(n) = \frac{n-1}{n} v(n-1) + \frac{1}{n} u(n) u^T(n) \frac{v(n-1)}{\|v(n-1)\|} (5)$$

Where $\frac{n-1}{n}$ the weight for the last estimates 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:

$$v(1) = \frac{1}{n} u(1) u^T(1) \frac{v(o)}{\|v(o)\|} (6)$$

Then, this vector will be the initial direction in the FastICA algorithm:
$w = v(1)$(7)

The FastICA algorithms will repeat until convergence the following rule:
$W_{new} = E[v(1).g(w^T.v(1))] - E[g'(w^T.v(1))].w,$(8)

Where $g'(x)$is the derivative of the function $g(x)$(10) it should be noted that this algorithm uses an approximation of negentropy inOrder to assure the non-Gaussianity of the independent vectors. Before starting the calculation of negentropy, a nonquadratic function G should be choosen, for example,

$$G(u) = -\exp\left(-\frac{u^2}{2}\right) \quad (9)$$

and its derivative:

$$g(u) = u.\exp\left(\frac{-u^2}{2}\right). \quad (10)$$

In general, the corresponding non-Gaussian vector w, for the estimated eigenvector $v(n)$, will be estimated using the following
repeated rule:

$$w_{new} = E\left[v(n).g\left(w^T.v(n)\right)\right] - E\left[g'\left(w^T.v(n)\right)\right].w. \quad (11)$$

B. Higher Order Non-Gaussian Vectors

The previous discussion only estimates the first non-Gaussian vector. One way to compute the other higher order vectors is 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 non-Gaussian 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 non- Gaussian vector, first the data is subtracted from its projection on the estimated first order eigenvector $v1(n)$, as shown in (12)

$$u_2(n) = u_{1(n)} - u_1^T(n)\frac{v_1(n)}{\|v_2(n)\|}\frac{v_2(n)}{\|v_2(n)\|} \quad (12)$$

where u1(n) = u(n). The obtained residual, u2(n), which is in thecomplementary space of v1(n), serves as the input data to theiteration step. In this way, the orthogonality is always enforcedwhen the convergence is reached, although not exactly so at earlystages.

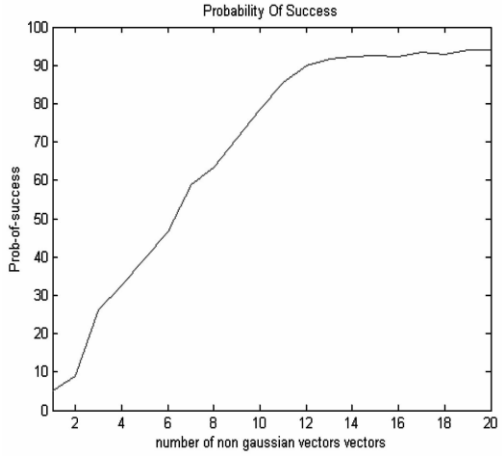


Fig. 3. Probability of success using UMIST Database as a function of number of non-Gaussian vectors

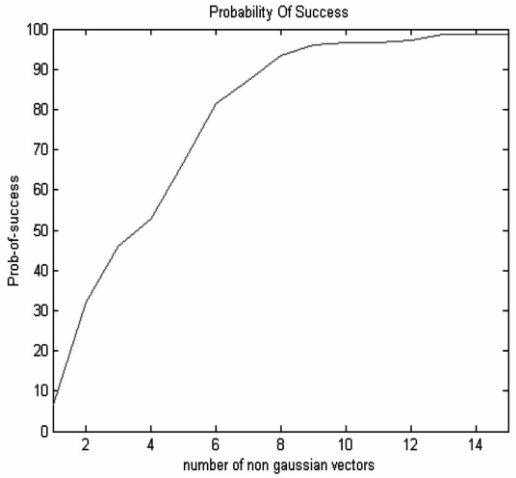


Fig. 4.Probability of success using Yale Database as a function of number of non-Gaussian vectors.

This, in effect, better uses the sample available and avoids the time-consuming GSO. After convergence, the non-Gaussian vector will also be enforced to be orthogonal, since they are estimated in complementary spaces. As a result, all the estimated vectors $W_k$will be: Non-Gaussian according to the learning rule in the algorithm. . Independent according to the complementary spaces introduced in the algorithm.

C. Algorithm Summary

Assume n different images $u(n)$ are given; let's calculate the first k dominant non-Gaussian vectors v_j(n). Assuming that $u(n)$ standsfor nth input image and v_j(n) stands for n^{th} update of the j^{th} non-Gaussian vector.
Combining IPCA and FastICA algorithms, the new algorithm can be summarized as follows:

For i = 1: n
img = input image from image data matrix;
u (i) = img;
for j=1:k
if j == i, initialize the jth non-Gaussian vector as:
$vj(i) = u(i);$
else

$$v_j(i) = \frac{i-1}{i}v_j(i-1) + \frac{1}{i}u(i)u^T(i)\frac{u_{j(i-1)}}{\|u_{j(i-1)}\|};$$

(vector update)

$$u(i) = u(i) - u^T(i)\frac{v_j(i)}{\|vj(i)\|}\frac{v_j(i)}{\|vj(i)\|};$$

(To ensure orthogonality)
end
$W = v_j(i);$

Repeat until convergence ($w_{new} = w$)

$W_{new} = E\ [vj(i).g(w^T.v_j(i)\ ] - E[g'(w^T.v_j(i))].w;$

(Searching for the direction that maximizes non-Gaussianity)

end

$v_j[i] = w^T.v_j[i];$ *(Projection on the direction of non-Gaussianity w)*

end

end

TABLE 1
Comparison between Different Algorithms

Methods	Success Rate using ORL	Success Rate using UMIST	Success Rate using Yale
PCA	60.88 %	75 %	94 %
Fisherface	88 %	94 %	96 %
DCV	No Data	No Data	97.3333 %
Batch PCA_ICA	79.75 %	77.2308 %	88.6667 %
IPCA_ICA	88.3724 %	94.3654 %	98.2424 %

D. Comparison with PCA-ICA Batch Algorithm

The major difference between the IPCA_ICA algorithm and the PCA_ICA batch algorithm is the real-time sequential process.IPCA_ICA doesn't need a large memory to store the whole data matrix that represents the incoming images. Thus, in each step, thisfunction deals with one incoming image in order to update the estimated non-Gaussian directions and the next incoming image canbe stored over the previous one. The first estimated non-Gaussian vectors (corresponding to the largest eigenvalues) in IPCAcorrespond to the vectors that carry the most efficient information. As a result, the processing of IPCA_ICA can be restricted to only aspecified number of first non-Gaussian directions. For example, the first 12 non-Gaussian vectors from 20 can construct an efficient basis with very low error rate for 380 faces of 20 persons in the UMIST database shown in Fig. 3, and the first nine non-Gaussian vectorsfrom 15 can construct an efficient basis for 150 faces of 20 persons in the Yale database shown in Fig. 4. 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 calculatingunwanted vectors. Also, ICA works usually in a batch mode where the extraction of independent components of the input eigenvectorscan be done only when these eigenvectors are present simultaneously at the input. It is very clear that from the time efficiencyconcern, IPCA_ICA will be more efficient and requires lessexecution time than PCA_ICA algorithm. Finally, IPCA_ICA givesa better face recognition performance than Batch PCA_ICA and thatis shown clearly in Table 1 by taking only a small number of basisvectors. These results are due to the fact that applying batch PCA onall the images will give the m noncorrelated basis vectors. ApplyingICA on the n out of these m vectors won't guarantee that theobtained vectors are the most efficient vectors. The basis vectorsobtained by the IPCA_ICA algorithm will have more efficiency orcontain more information than those chosen by the batch algorithm.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

1. Face Recognition Evaluated by Nearest Neighbor Algorithm

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

$$c = \frac{v_{test} \cdot v_{train}}{\|v_{test}\| \|v_{train}\|}. \qquad (13)$$

Each Face Database was truncated into two sets. The training set that contains images used to calculate the independent non- Gaussian 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 thecoordinates of each image with respect to the basis vtrain. Each new testing image vtest is compared to whole set of training images vtrain in order to come up with nearest one that corresponds to the maximum cosine c in (13).

2. Face Recognition Performance

Three popular face databases were used to demonstrate the effectiveness of the proposed IPCA_ICA algorithm.The ORL [11] 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/noglasses).

All persons are in the up-right, frontal position, with tolerance for some side movement. The UMIST [12] was taken from the University of Manchester Institute of Science and Technology. It is a multiview database, consisting of 575 images of 20 people, each covering a wide range of poses from profile to frontal views. The Yale [13] was 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 the following total expressions or configurations: center-light, with glasses, happy, left-light, without glasses, normal, right-light, sad, sleepy, surprised, and wink. Each image in the ORL database is scaled into (92 × 112), in the UMIST Database is scaled into (112 ×_ 92), and in the Yale Database is cropped and scaled into (126 × 152). 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. The partition of the ORL database into training and testing sets is done following the recommendation of [14], which call for six images per person randomly chosen for training, and the other four for testing. Thus, a training set of 240 images and a test set with 160 images are created. For the UMIST, 19 poses per person will be taken to form a new database of 380 images. Thus, six images per person are randomly chosen toproduce a training set of 120 images. The remaining 260 images are used to form the test set. The partition of the Yale database is done following the recommendation of [15]. Thus, 10 samples per subject are obtained yielding a face database size of 150. The recognition rates were computed by the "leave-one-out" strategy [16] since thetraining set size is relatively small. In the following experiments on the ORL and the UMIST databases, the figures of merit are success rates averaged over four runs for the UMIST database and three runs for ORL database,each run being performed on such random partitions in the two databases. In the other experiment using the Yale database, the success rate will be made average over 10 runs (each run corresponds to a different test image).The IPCA_ICA algorithm is compared against four feature selection methods, namely, Discriminative Common Vectors algorithm (DCV) [15], the PCA algorithm [10], the Fisherfaces algorithm [17], and the Batch PCA_ICA. For each of the three methods, the face recognition procedure consists of: 1) a feature extraction step where two kinds of feature representation of each training or test sample are extracted by projecting the sample onto the two feature spaces generalized by the PCA, the Fisherface, respectively, and 2) a classification step in which each feature representation obtained in the first step is fed into a simple nearest neighbor classifier. It shouldbe noted at this point that, since the focus in this paper is on feature extraction, a very simple classifier, namely, nearest neighbor, is used in Step 2). IPCA_ICA is compared to the DCV algorithm using only the Yale database because of the availability of results on that database compared to others used in this paper. In addition, IPCA_PCA is compared to batch FastICA algorithm that applies PCA and ICA one after the other. In FastICA, the reduced number of eigenvectors obtained by PCA batch algorithm is used as input vectors to ICA batch algorithm in order to generate the independent non-Gaussian vectors. Here, FastICA process is not a real-time process because the batch PCA requires a previous calculation of covariance matrix before processing and calculating the eigenvectors.Notice here that PCA, Batch PCA_ICA, and PCA_ICA algorithms are experimented using the same method of introducing and inputting the training images and tested using also the same nearest neighbor procedure. However, the DCV and the Fisherface results are obtained without testing from other papers [15], [17]. The success rates of the IPCA_PCA algorithm of three different runs on the ORL database indicate that the average success rate ofthis algorithm is 88.3724 percent. The results using the UMIST database indicate that the average success rate of this algorithm is 94.3654 percent. In addition, the 10 different runs on the Yale database indicate that the average success rate is 98.2424 percent. A comparison between the performances of these algorithms using the threedatabases is shown in Tables 1 and 2. Figs. 3 and 4 show two plots of the probability of success of the algorithm as a function of number of independent non-Gaussian vectors using the UMIST and the Yale databases. Assuming thatthe maximum number of independent non-Gaussian vectors is the total number of different persons in a database, the compression can be achieved by reducing that number. For the UMIST database, the maximum number of non-Gaussian directions is 20. Fig. 3 shows that choosing only 14

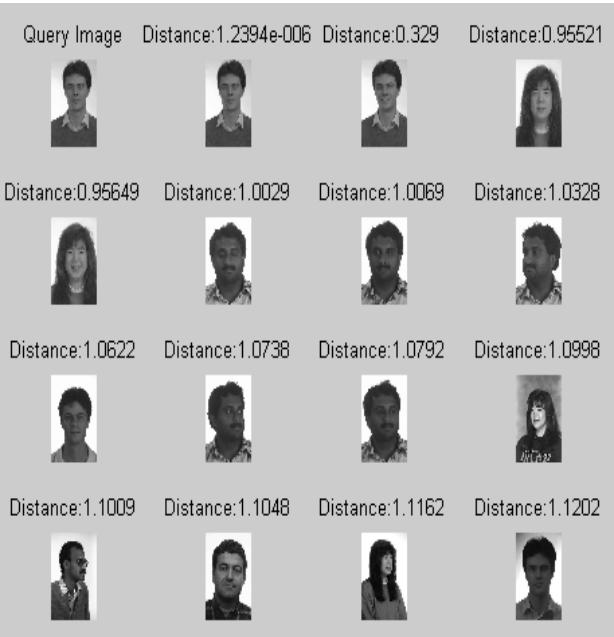
TABLE 2
Average Success Rate for All Used Image Databases

Methods	Average Success Rate
PCA	76.62 %
Fisherface	92.67 %
Batch PCA_ICA	81.88 %
IPCA_ICA	93.66 %

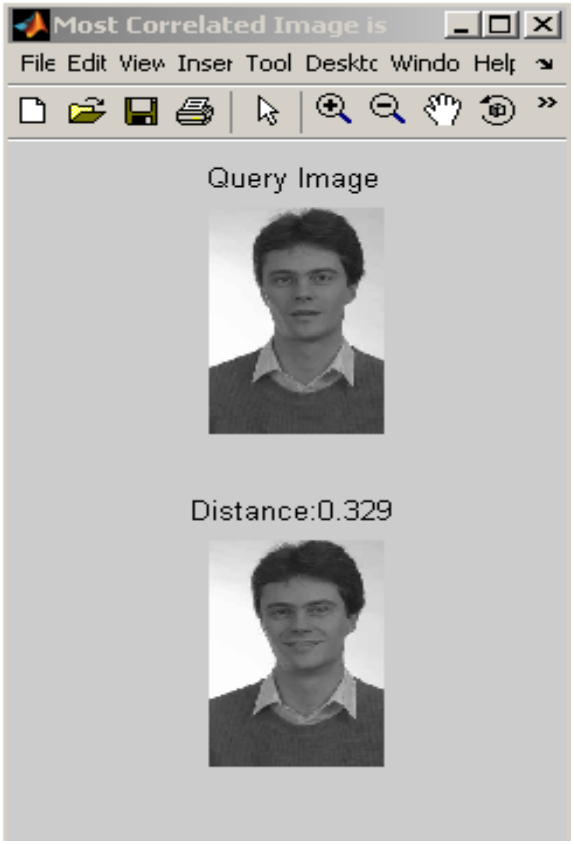
non-Gaussian vectors will give approximately the same performance. For the Yale database, Fig. 4 shows that eight non-Gaussian vectors will give approximately thesame performance as choosing all the 15 vectors. It should be noted that for the ORL database the total number ofimages is 400 and the maximum number of non-Gaussian vectors is 20. Let's define sample-to-dimension ratio as n/d, where n is the number of samples and d is the dimension of the sample space. The IPCA_PCA algorithm gives better performance when the training set size becomes large because the lower the sample-todimension ratio the harder a statistical estimation problem becomes. Usually, dealing with images introduces a very low sample-to-dimension ratio because of the big number of pixels in each image. Using this algorithm, Fig. 5 shows the first 10 independent non-Gaussian vectors using the three databases.

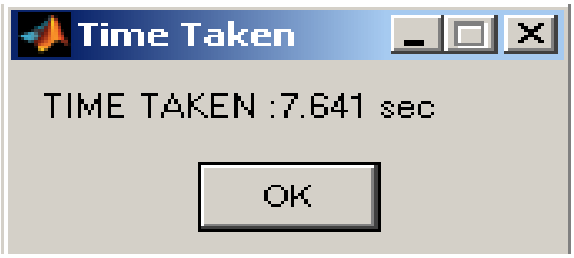
V. CONCLUSION AND FUTURE WORK

In this paper, a new feature extraction method for face recognition tasks based on incremental update of the non-Gaussian independent vectors has been proposed. The method concentrates on a challenging issue of computing dominating non-Gaussian 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 andidentifying) in high speed and obtain good results.Its effectiveness and good performance has been proven by experiments. The proposed IPCA_ICA 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 Eigenface, the Fisherface, and the DCV.In Table 2, it is clear that IPCA_ICA achieves higher average success rate than the Eigenface, the Fisherface, and the FastICA



Total Elapsed Time Figure 5(a)





Total Elapsed Time Figure 5(b)

Total Elapsed Time Figure 5(c)

methods. Table 1 shows that this algorithm has a better success rate than the DCV method using theYale Database. The importance of the result presented here is potentially beyond the apparent technical scope interesting to the computer vision community. Analyzing human brain states that what but, more importantly and more fundamentally, developing the computing engine itself,from real-world, online sensory data streams. Although a lot of studies remain to be done and many open questions are waiting to be answered. What is the relationship between this algorithm and our brain? A clear answer is not available yet, but Rubner and Schulten [18] have proven that the well-known mechanisms of biological Hebbian learning and lateral inhibition between nearby neurons [19, p. 1,020 and p. 376] result in an incremental way ofcomputing PCA. The proposed method can be seen as a small description of the developmental mechanisms of our brain.

REFERENCES

[1] J. Karhunen and J. Joutsensalo, "Representation and Separation of SignalsUsing Non Linear PCA Type Learning," Neural Networks, vol. 7, no. 1, 1994.

[2] P. Common, "Independent Component Analysis, a New Concept?" SignalProcessing, vol. 36, no. 3, 1994.

[3] J.J. Atick and A.N. Redlich, "What Does the Retina Know about NaturalScenes?" Neural Computing, vol. 4, pp. 196-210, 1992.

[4] J.F. Cardoso, "Blind Signal Separation, Statistical Principles," Proc. IEEE,vol. 9, no. 10, 1998.

[5] H. Murase and S.K. Nayar, "Visual Learning and Recognition of 3-DObjects from Appearance," Int'l J. Computer Vision, vol. 14, no. 1, pp. 5-24, Jan 1995

[6] Y. Cui and J. Weng, "Appearance-Base Hand Sign Recognition fromIntensity Image Sequences," Computer Vision and Image Understanding,vol. 78, pp. 157-176, 2000.

[7] S. Chen and J. Weng, "State-Based SHOSLIF for Indoor Visual Navigation,"IEEE Trans. Neural Networks, vol. 11, no. 6, pp. 1300-1314, 2000.

[8] Proc. NSF/DARPA Workshop Development and Learning, J. Weng andI. Stockman, eds., Apr. 2000.

[9] E.P. Simoncelli, "Higher Order Statistical Models of Visual Images," Proc.IEEE Workshop Higher Order Statistics, June 1999.

[10] M. Turk and A. Pentland, "Eigenfaces for Recognition," J. CognitiveNeuroscience, vol. 3, no. 1, pp. 71-86, 1991.

[11]"ORL Face Database," AT&T Laboratories Cambridge, http://www.camorl.co.uk/facedatabase.html, 2005.

[12]"UMIST Face Database," Daniel Graham, http://images.ee.umist.ac.uk/danny/database.html, 2005.

[13]"Yale Face Database," Columbia University, http://www1.cs.columbia.edu/belhumeur/pub/images/yalefaces/, 2005.

[14] Y. Wang, T. Tan, and Y. Zhu, "Face Verification Based on Singular ValueDecomposition and Radial Basis Function Neural Network+," Inst.Automation, Chinese Academy of Sciences, Beijing, P.R. China, 100080.2000.

[15] H. Cevikalp, M. Neamtu, M. Wilkes, and A. Barkana, "DiscriminativeCommon Vectors for Face Recognition," IEEE Trans. Pattern Analysis andMachine Intelligence, vol. 27, no. 1, pp. 6-9, Jan. 2005.

[16] K. Fukunaga, Introduction to Statistical Pattern Recognition, second ed. NewYork: Academic Press, pp. 831-836, Aug. 1996.

[17] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman, "EigenfacesvsFisherfaces: Recognition Using Class Specific Linear Projection," IEEETrans. Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 711-720,July 1997.

[18] J. Rubner and K. Schulten, "Development of Feature Detectors by Self-Organization," Biological Cybernetics, vol. 62, pp. 193-199, 1990.

[19] Principles of Neural Science, E.R. Kandel, J.H. Schwartz, and T.M. Jessell,eds., third ed. Norwalk, Conn.: Appleton and Lange, 1991.

[20] J. Haddadnia, K. Faez, and A. Majid, "N-FeatureNeural Network Human Face Recognition," ImageVision Computing 22, pp. 1071-1082, 2004.

[21] B.L. Zahng, H. Zhang, and S.S. Ge, "Face Recognitionby Applying Wavelet Subband Representation andKernel Associative Memory," IEEE Trans. NeuralNetworks, vol. 15, no. 1, pp. 166-177, Jan. 2004.

[22] A. F. Abate, M. Nappi, D. Riccio, and G. Sabatino,"2D and 3D Face Recognition: A Survey," PatternRecognition Letters, 28, pp. 1885-1906, 2007.