Appearance-based Statistical Methods for Face Recognition

Kresimir Delac [1], Mislav Grgic [2], Panos Liatsis [3]
[1] Croatian Telecom, Savska 32, Zagreb, CROATIA
[2] University of Zagreb, Faculty of EE & Comp, Unska 3/XII, Zagreb, CROATIA
[3] Information and Biomedical Engineering Centre, City University London, UNITED KINGDOM
E-mail: *kdelac@ieee.org*

Abstract - *Different statistical methods for face recognition have been proposed in recent years. They mostly differ in the type of projection and distance measure used. The aim of this paper is to give an overview of most popular statistical subspace methods for face recognition task. Theoretical aspects of three algorithms will be considered and some reported performance evaluations will be given.*

Keywords - *Face Recognition, PCA, ICA, LDA, Subspace Analysis*

1. INTRODUCTION

Face recognition has gained much attention in recent years and has become one of the most successful applications of image analysis and understanding. Face recognition conferences are emerging and sophisticated commercial systems have been developed that achieve rather high recognition rates. A general statement of the problem can be formulated as follows [1]: Given still or video images of a scene, identify or verify one or more persons in the scene using a stored dataase of faces. This area of research is important not only because of the applications in human-computer interaction, biometrics and security, but also because it is a typical, pattern recognition problem that, if successfully solved, could help solve other pattern classification problems.

The first approach used for recognizing faces (and the most intuitive one) was correlation, but all such methods were computationally expensive so it was only natural to pursue dimensionality reduction schemes.

In this paper, three appearance-based statistical methods, namely Principal Component Analysis (PCA), Independent Component Analysis (ICA) and Linear Discriminant Analysis (LDA), are described. PCA [2], [3], [4] is a subspace projection technique widely used for face recognition. It finds a set of representative projection vectors such that the projected samples retain most information about original samples. The most representative vectors are the eigenvectors corresponding to the largest eigenvalues of the covariance matrix. While PCA deals with variance (second-order statistics), ICA [5] captures both second and higher-order statistics and projects the input data onto the basis vectors that are as statistically independent as possible. We can state that ICA is a generalization of PCA. LDA [6], [7], unlike PCA or ICA, uses the class information and finds a set of vectors that maximizes Fisher Discriminant Criterion. It simultaneously maximizes the between-class scatter while minimizing the within-class scatter in the projective feature vector space. While PCA and ICA can be called unsupervised learning techniques, LDA is supervised learning technique because it needs class information for each image in the training process.

When face recognition was at its beginning each research group collected their own database of images (e.g. Harvard, USC etc). Later, there emerged a need for a uniform benchmark database and thus FERET database was collected at NIST (National Institute of Standards and Technology) [8] and became the most used face database for testing race recognition algorithms. The images were collected between 1993 and 1996. The proposed gallery set for frontal face recognition consists of images of 1,196 individuals and probe images are divided into four sets, namely *fb*, *fc*, *dupI* and *dupII* set (for details please refer to [8]) However, many authors agree that FERET database favors one sort of algorithms so there are other databases that are comparable to FERET and are often used for testing today (Yale [9], AR [10], XM2TVS [11], CMU PIE [12]). They are collected for specific purposes and so the AR database contains occlusions due to eye glasses and scarf and the CMU PIE database is collected with well-constrained pose, illumination and expression. The Yale database contains 160 frontal face images covering sixteen individuals taken under ten different conditions (different illumination and expression). The XM2TVS database is especially designed for multi-modal biometrics, including audio and video cues and is not available free of charge.

The rest of this paper is organized as follows: in Section II the idea of face space in comparison to image space is given, Section III deals with linear subspace analysis describing three most popular methods in that area, in Section IV performance comparison of described methods (as reported by other research groups) is given.

2. FACE SPACE

Generally, a two dimensional image $\Gamma(x,y)$ of size m-by-n pixels can be viewed as a vector (or a point) in high dimensional space. The easiest way to create a vector from an array is to concatenate its columns, thus getting a vector $\boldsymbol{X} = [x_1 \ldots x_N]^T$, where $N = m \times n$. Each pixel of the image then corresponds to a coordinate in N-dimensional space. We will refer to this space as *image space*. Such a space has huge dimensionality ($\Re^N$) and recognition there would be computationally inefficient.

However, if an image of an object is a point in image space, a collection of M images of the same sort of an object represents a set of points in the same *subspace* of the original image space. These points may be considered as samples of probability distribution. Theoretically, all possible images of one particular object define a lower-dimensional (possibly disconnected) *manifold*, embedded within the high-dimensional image space. For face recognition purposes we refer to this as *face space*. Its intrinsic dimensionality is determined by the number of degrees of freedom within face space. Appearance-based object recognition (i.e. subspace analysis) deals with the following questions [2]: what is the relationship between points in image space that correspond to all images of a particular object (face)? Is it possible to efficiently characterize this subset of all possible images? Can this subset be learned from a set of training images? What is the "shape" of this subset?

Basically, the goal of subspace analysis is to determine the intrinsic dimensionality and to extract the principal modes (*basis functions*) of the principal manifold. By doing this in a subspace, compression is achieved (computational efficiency), data samples are drawn from a normal distribution (meaning that axes of large variance probably correspond to data while axes of small variance are probably noise) and, because data will be mean centered, Euclidian distance in subspace is inversely proportional to correlation between source images.

3. LINEAR (SUBSPACE) ANALYSIS

In the following sections three classical linear appearance-based classifiers (PCA, ICA and LDA) will be described. Each of these has its own set of basis functions which are derived based on different statistical viewpoints. After deriving basis vectors, a face image is projected onto them and the projection coefficients are used as the feature representation of each face image. The matching score between the test image and each training image is calculated between their coefficients vectors where the largest value represents the recognized object. The necessary assumption for all these classifiers is that the principal manifold is *linear*.

3.1. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) [3], [4] is a method to efficiently represent a collection of sample points, reducing the dimensionality of the description by projecting the points onto the principal axes, where an orthonormal set of axes points in the direction of maximum covariance in the data [2]. These vectors best account for the distribution of face images within the entire image space. PCA minimizes the mean squared projection error for a given number of dimensions, and provides a measure of importance (in terms of total projection error) for each axis.

PCA is closely related to popular signal processing technique known as the Karhunen-Loeve transform (KLT). It can in fact be shown that under the assumption that the data is zero-mean the formulations of PCA and KLT are identical [13].

Let us now describe the PCA algorithm as proposed in [3]. First we will *create the eigenspace*. This step is the initialization of the system. Let the training set of M face images be $\Gamma_1, \Gamma_2, \ldots, \Gamma_M$. The average face of the set is defined by:

$$\Psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n \tag{1}$$

Each face differs from the average face by the vector $\boldsymbol{\Phi}_i = \boldsymbol{\Gamma}_i - \boldsymbol{\Psi}$, where $i = 1$ to M. We shall rearrange these vectors in a matrix $A = [\boldsymbol{\Phi}_1, \ldots, \boldsymbol{\Phi}_M]$ of dimension $N \times M$, which will then be subject to PCA. Matrix A has zero-mean (mean value subtracted) vectors of each training face image in its columns. What we have just done is in fact a translation of the origin to the *mean* face (see Fig. 1. for the illustration of the mean face).

The next goal is to find a set of $M - 1$ orthogonal vectors, $\boldsymbol{e}_i$, which best describes the distribution of the input data in a least-squares sense, i.e., the Euclidian projection error is minimized. We start by finding the covariation matrix:

$$C = A \cdot A^T \tag{2}$$

and then we use eigenvector decomposition:

$$C \cdot e_i = \lambda_i \cdot e_i \tag{3}$$

where $\boldsymbol{e}_i$ and λ_i are eigenvectors and eigenvalues of covariation matrix C, respectively. We can do this because C is real and symmetric. λ is a diagonal matrix with eigenvalues on its main diagonal.

Fig. 1. *Mean face* calculated from 1,196 FERET gallery images [8].

Once the eigenvectors of C are found, they are sorted according to their corresponding eigenvalues. Larger eigenvalue means that associated eigenvector captures more of the data variance. The efficiency of the PCA approach comes from the fact that we can eliminate all but the best k eigenvectors (with the highest k eigenvalues). Since PCA assumes the directions with the largest variances are the most *principal* (important), these eigenvectors will then span the M' dimensional face space and that is the new feature space for recognition. Eliminating eigenvectors associated with small eigenvalues actually eliminates the noise from the image.

There are at least three proposed ways to eliminate eigenvectors. First is the mentioned elimination of eigenvalues with smallest eigenvalues. This can be accomplished by discarding the last 60% of total number of eigenvectors. The second way is to use the minimum number of eigenvectors to guarantee that energy E is greater than a threshold. A typical threshold is 0.9 (90% of total energy). If we define E_i as the energy of the ith eigenvector, it is the ratio of the sum of all eigenvalues up to and including i over the sum of all the eigenvalues:

$$E_i = \frac{\sum_{j=1}^{i} \lambda_j}{\sum_{j=1}^{k} \lambda_j} \tag{4}$$

where k is the total number of eigenvectors (Fig. 2.).

The third variation depends upon the stretching dimension. The stretch for the ith eigenvector is the ratio of that eigenvalue over the largest eigenvalue (λ_1):

$$s_i = \frac{\lambda_i}{\lambda_1} \tag{5}$$

All eigenvectors with s_i greater than a threshold are retained. A typical threshold is 0.01. Some authors also discard the first few eigenvectors because they seem to capture mainly the lighting variations (this can be confirmed by looking at the first two faces at the top row of Fig. 5.). However, it is rather questionable if this last step actually improves recognition rate.

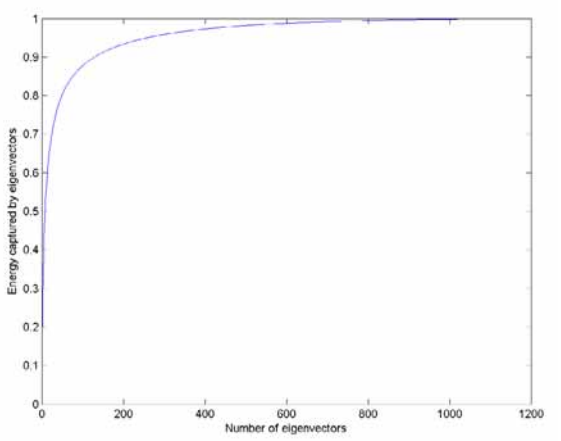

Fig. 2. Energy captured by retaining the number of eigenvectors with the largest eigenvalues, calculated from 1,196 FERET gallery images [8]. It is clearly seen that retaining only 200 eigenvectors (of total 1,196 vectors) captures more then 90% of the energy.

Each eigenvector has the same dimensionality as a face image and looks as a sort of a "ghost" face (if rearranged and viewed as a picture), so we call them *eigenfaces* (Fig. 5, top row). Transforming a point to a new space is a linear transformation so eigenvectors are merely linear combinations of the training images. The last step is to calculate the average face image for each individual (if there is more than one instance of that individual) and to project this image into the face space as the individual's class prototype. Ideally, two images of the same person should project to the same point in eigenspace. Any difference between the points is unwanted variation. Two images of different subjects should project to points that are as far apart as possible. This is the main idea behind the recognition in subspaces.

After creating the eigenspace we can proceed to *recognition using eigenfaces*. Given a new image of an individual Γ, the pixels are concatenated the same way as the training images were, the mean image Ψ is subtracted and the result is projected into the face space:

$$\omega_k = e_k^T (\Gamma - \Psi) \tag{6}$$

for k = 1, ... ,M'. These calculated values of ω together form a vector $\boldsymbol{\Omega}^T = [\omega_1\ \omega_2\ \ldots\ \omega_{M'}]$ that describes the contribution of each eigenface in

representing the input face image. In fact, this is the projection of an unknown face into the face space. $\boldsymbol{\Omega}$ is then used to establish which of the pre-defined face classes best describes the new face. The simplest way to determine which face class provides the best description of the input face image is to find the face class k that minimizes the Euclidian distance:

$$\varepsilon_k = \sqrt{\|\Omega - \Omega_k\|^2} \tag{7}$$

where $\boldsymbol{\Omega}_k$ is a vector describing the kth face class. A face is classified as belonging to a certain class when the minimum ε_k (i.e. the maximum matching score) is below some certain threshold. Besides Euclidian distance other similarity measures or metrics can be used, such as L1 norm (also called the City Block Distance):

$$d(x,y) = |x - y| = \sum_{i=1}^{k} |x_i - y_i|, \tag{8}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are any two vectors, or Mahalanobis distance:

$$d(x,y) = -\sum_{i=1}^{k} \frac{1}{\sqrt{\lambda_i}} \cdot x_i \cdot y_i \tag{9}$$

where λ_i is the ith eigenvalue corresponding to the ith eigenvector.

There is one important property of PCA that needs to me mentioned. In order for PCA to work one must assume that mean and variance are sufficient statistics to entirely describe the data. The only zero-mean probability distribution that is fully described by the variance is the Gaussian distribution. In the next section we shall describe an algorithm that works even if the distribution of data is not Gaussian. In practice though, quite a lot of the real world data are Gaussian distributed (thanks to the Central Limit Theorem) and PCA thus represents good means to roughly describe the data.

3.2. Independent Component Analysis (ICA)

As seen in the previous section, PCA makes one important assumption: the probability distribution of input data must be Gaussian. When this assumption holds, covariance matrix contains all the information of (zero-mean) variables. Basically, PCA is only concerned with second-order (variance) statistics. The mentioned assumption need not be true. If we presume that face images have more general distribution of probability density functions along each dimension, the representation problem has more degrees of freedom. In that case PCA would fail because the largest variances would not correspond to meaningful axes of PCA.

Independent Component Analysis (ICA) [5] minimizes both second-order and higher-order dependencies in the input. It keeps the assumption of linearity but abandons all other that PCA uses.

Although the amplitude spectrum is captured by second-order statistics in PCA, there remains the phase spectrum that lies in higher-order statistics. It is believed that this high-order statistics (i.e. the phase spectrum) contains the structural information in images that drives human perception [5].

ICA attempts to find the basis along which the data (when projected onto them) are – *statistically independent*. Mathematically, if (x,y) are two independent components (bases), then:

$$P[x,y] = P[x] \cdot P[y] \tag{10}$$

where $P[x]$ and $P[y]$ are distributions along x and y and $P[x,y]$ is the joint distribution. To put it simply, ICA is a way of finding a linear non-orthogonal coordinate system in any multivariate data. The directions of axes of this coordinate system are determined by both the second and higher order statistics of the original data. The goal is to perform a linear transform, which makes the resulting variables as statistically independent from each other as possible. It is a generalization of PCA (so PCA can be derived as a special case of ICA). It is closely related to *blind source separation* (BSS) problem (Fig. 3.), where the goal is to decompose an observed signal into a linear combination of unknown independent signals [14].

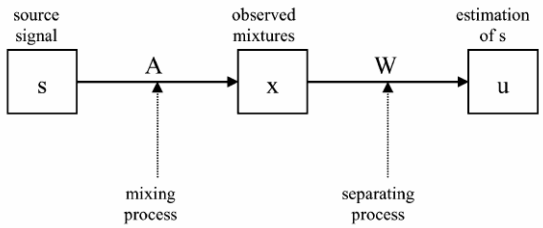

Fig. 3. Blind source separation model.

Let $\boldsymbol{s}$ be the vector of unknown source signals and $\boldsymbol{x}$ be the vector of observed mixtures. If A is the unknown mixing matrix, then the mixing model is written as $\boldsymbol{x} = A \cdot \boldsymbol{s}$, where source signals are independent of each other and A is invertible. ICA tries to find the mixing matrix A or the separating matrix W such that

$$U = W \cdot x = W \cdot A \cdot s \tag{11}$$

is the estimation of the independent source signals [13]. There are many algorithms that perform ICA (InfoMax [13], JADE [15], FastICA [16]) but they all seem to converge to the same solution for any given data set. Their main principle is to iteratively optimize a smoothing function whose global optima occurs when the output vectors U are independent.

ICA can be used in face recognition in two different ways [5]. It is standard practice to refer to them as *Architecture I* and *Architecture II* and this nomenclature will be adopted here as well. Their basic differences can be seen in Fig. 4.

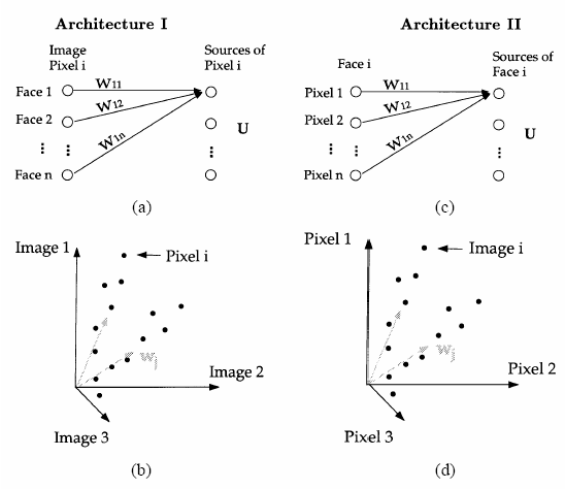

Fig. 4. Two architectures for performing ICA on faces [5]. (a) Performing source separation on face images produced independent component images in the rows of *U*. (b) The gray values at pixel location i are plotted for each image. ICA *Architecture I* finds weight vectors in the directions of statistical dependencies among the *pixel locations*. (c) *U* has factorial code in its columns. (d) ICA *Architecture II* finds weight vectors in the directions of statistical dependencies among the *face images*.

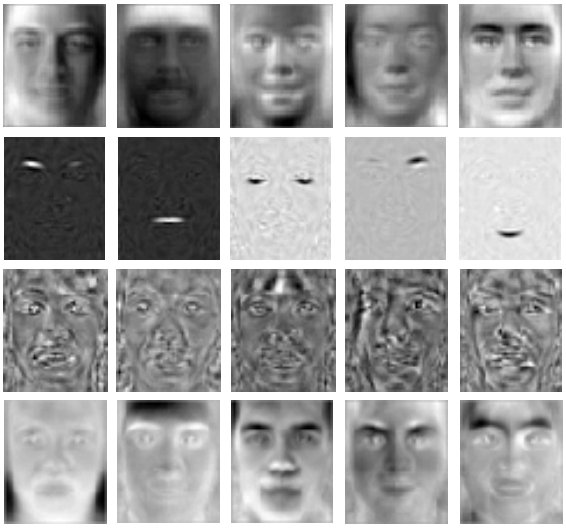

Fig. 5. An illustrative example of differences between PCA, ICA *Architecture I,* ICA *Architecture II* and LDA. Top row shows top eight PCA eigenfaces. The second row shows localized feature vectors for ICA *Architecture I*. The third row shows eight non-localized ICA feature vectors for ICA *Architecture II*. Bottom row shows LDA representation vectors (*Fisherfaces*).

In *Architecture I* [5], [14] the input face images X are considered to be a linear mixture of statistically independent basis images S combined by an unknown matrix A. Each row vector of X is a different image. ICA learns the weights matrix W (Fig. 4. (a), (b)) such that the rows of $U = W \cdot X$ are as statistically independent as possible. In this architecture, the face images are variables and pixel values are observations. The source separation is performed in face space and the source images estimated by the rows of U are then used as basis images to represent faces. The compressed representation of a face image is a vector of coefficients used for linearly combining the independent basis images to generate the image (much like the PCA). Eight sample basis images (rows of U, each one rearranged to original image format) derived this way can be seen in the second row of Fig. 5. Notice the spatial localization, unlike the PCA (top row) or *Architecture II* (bottom row). The following conclusion can be drawn from this example: each row of the mixing matrix W found by ICA represents a cluster of pixels that have similar behavior across images. We say that *Architecture I* produces *statistically independent basis images*.

Although the basis images obtained in *Architecture I* are approximately independent, the coefficients that code each face are not necessarily independent. In *Architecture II* [5], [14], the goal is to find statistically independent coefficients for input data. The rows of data matrix X are now different pixels and the columns are different images. The pixels are now variables and the images are observations (Fig. 4. (c), (d)). The source separation is performed on pixels and each row of the learned weight matrix W is an image. A (inverse matrix of W) contains the basis images in its columns. The statistically independent source coefficients in S that comprise the input images are recovered in the columns of U. Eight sample basis images derived this way can be seen in the third row of Fig. 5. In this approach, each column of the mixing matrix W^{-1} found by ICA attempts to "get close to a cluster of images that look similar across pixels". This way, *Architecture II* tends to generate basis images that are even more face-like than the one derived by PCA. In fact, the basis found by ICA will average only images that look alike. We say that *Architecture II* produces *statistically independent coefficients* (it is sometimes called *factorial code method* as well).

If training data for face recognition system would have 500 images, ICA algorithm would try to separate 500 independent components, which has high computational complexity, if not impossible. That is why it is common practice to perform ICA on the PCA coefficients (rather then directly on the input images) to reduce the dimensionality [5].

Face recognition using ICA can be summarized by the following: compare the test image independent components with the independent

components of each training image by using a similarity measure. The result (the recognized face) is the training image, which is the closest to the test image. Similarity measure used in [5] was the nearest neighbor algorithm with cosine similarity. Let b denote the coefficient vector. Coefficient vectors in each test set were assigned the class label of the coefficient vector in the training set that was most similar as evaluated by the cosine of the angle between them:

$$c = \frac{b_{test} \cdot b_{train}}{\|b_{test}\| \cdot \|b_{train}\|} \tag{12}$$

Comparison of reported performance will be given in Section IV.

3.3. Linear Discriminant Analysis (LDA)

Both PCA and ICA do not use face class (category) information. The training data is taken as a whole. Linear Discriminant Analysis (LDA) finds an efficient way to represent the face vector space by exploiting the class information [6], [7]. It differentiates individual faces but recognizes faces of the same individual. LDA is often referred to as a Fisher's Linear Discriminant (FLD).

The images in the training set are divided into the corresponding classes. LDA then finds a set of vectors W_{LDA} such that Fisher Discriminant Criterion is maximized:

$$W_{LDA} = \arg\max_{W} \frac{W^T \cdot S_B \cdot W}{W^T \cdot S_W \cdot W} \tag{13}$$

where S_B is the between-class scatter matrix and S_W is the within-class scatter matrix, defined by:

$$S_B = \sum_{i=1}^{c} N_i \cdot (x_i - \mu) \cdot (x_i - \mu)^T \tag{14}$$

$$S_W = \sum_{i=1}^{c} \sum_{x_k \in X_i} (x_k - \mu_i) \cdot (x_k - \mu_i)^T \tag{15}$$

where N_i is the number of training samples in class i, c is the number of distinct classes, μ_i is the mean vector of samples belonging to class i and X_i represents the set of samples belonging to class i. S_W represents the scatter of features around the mean of each face class and S_B represents the scatter of features around the overall mean for all face classes. These vectors, the same as the PCA vectors, if rearranged are very face-like, so they are often called *Fisherfaces* (Fig. 5, bottom row).

It is not difficult to demonstrate that the solution of the maximization problem of (13) is the solution of generalized eigensystem:

$$S_B \cdot V = \Lambda \cdot S_W \cdot V \tag{16}$$

where V is the eigenvector (fisherfaces) matrix and Λ are the corresponding eigenvalues of the within-class and between-class scatter matrices. This system is easily solved if written like this:

$$S_W^{-1} \cdot S_B \cdot V = \Lambda \cdot V \tag{17}$$

This approach can produce some problems. Let us state some of them: 1) this eigensystem does not have orthogonal eigenvectors because $S_W^{-1} \cdot S_B$ is, in general, not symmetric, 2) matrices S_B i S_W are usually too big, 3) S_W could be singular and then noninvertible. All these problems can be bypassed by using the PCA decomposition previous to LDA [6]. However, the system in (13) will then give reduced eigenvectors v, that need to be transformed into true eigenvectors V using $V_{FF}=v \cdot V_{EF}$, where V_{EF} and V_{FF} are the PCA and fisher projection matrices, respectively [17].

After the eigenvectors have been found (and only the ones corresponding to largest eigenvalues have been kept), the original images are projected onto them by calculating the dot product of the image with each of the eigenvectors. Recognition is again done by calculating the distance of the projected input image to all the training images projections, and the nearest neighbor is the match.

4. PERFORMANCE COMPARISON

Let us now make a comparison between these methods. In all three algorithms, classification is performed by first projecting the input images into a subspace via a projection (basis) matrix and then comparing the projection coefficient vector of the input to all the pre-stored projection vectors or labeled classes to determine the input class label.

Various groups have reported various performance results for these three algorithms over the years and no straightforward conclusion can be easily drawn.

Zhao *et al* [18] report improved performance for combined PCA and LDA approach over the pure LDA. In [6] four methods were compared (correlation, a variant of linear subspace method, PCA and PCA+LDA). The results showed that PCA+LDA performed significantly better than the other three methods. In [17] authors state that LDA with cosine distance measure outperformed all other tested systems, claiming that LDA dimensionality reduction works better than other projection methods. However, Beveridge *et al* [19] state that on their tests using different distance measures the LDA algorithm performed uniformly worse than PCA, but they do not give any explanation as to why it is so. Martinez *et al* [7] argued that for a small training set (two images per class) PCA can outperform LDA, but this is not the case for a larger training set.

Table 1. Results reported by different research groups testing three described algorithms

Research group	Database	Algorithms tested	Best result
Zhao [18]	FERET & USC	LDA, PCA+LDA	PCA+LDA
Belhemur [6]	Harvard & Yale	Correlation, Linear Subspace, PCA, LDA	LDA
Navarrete [17]	FERET & Yale	PCA, LDA, EP	LDA
Beveridge [19]	FERET	PCA, PCA+LDA	PCA
Bartlett [5]	FERET	PCA, ICA	ICA
Baek [20]	FERET	PCA, ICA	PCA
Liu [21]	FERET	PCA, LDA, ICA	ICA

Comparing PCA and ICA, Bartlett *et al* [5] report that both ICA representations (*Architecture I* and *II*) outperformed the pure PCA, for recognizing images of faces sampled on a different day from the training images. A classifier that combined two representations outperformed PCA on all test sets. However, the work done in [20] clearly contradicts the one in [5]. Liu *et al* [21] suggest that for enhanced performance ICA should be carried out in a compressed and whitened space where most of the representative information of the original data is preserved and the small eigenvalues discarded. The dimensionality of the compressed subspace is decided based on the eigenvalue spectrum from the training data. The discriminant analysis shows that the ICA criterion, when carried out in the properly compressed and whitened space, performs better than the eigenfaces and Fisherfaces methods, but its performance deteriorates significantly when augmented by an additional discriminant criteria such as the FLD. Many other authors claim that PCA (or some of its modifications) by far outperforms both ICA and LDA.

5. CONCLUSION

As seen in the previous section, no straightforward conclusion can be drawn on overall performance results of three described algorithms. At best, we can state that each of these algorithms performs best for a specific task.

However, we believe that there are not enough independently conducted comparisons of these three algorithms, performed under the same initial conditions (i.e. the same preprocessing – image rotation, cropping, enhancement). Furthermore, never are all possible implementations considered (various projection methods combined with various distance measures). Our further work will consist of implementing the described algorithms in Matlab® and testing them using as many as possible different similarity measures, to try to produce the best combination for a specific task. We expect to face some difficulties in directly comparing LDA to other two algorithms because of its need for more than one image of an individual, so we will have to think of the methodology to overcome that issue (following the lead of [19] and [20] perhaps). Also, a fair comparison technique should be designed regarding the dimensionality of recognition and intermediate subspaces and algorithm performance. Perhaps the best way is to test couple of possible dimensionalities and to report and compare only the best results for a specific algorithm given the same task.

ACKNOWLEDGMENT

Portions of the research in this paper use the Color FERET database of facial images collected under the FERET program.

REFERENCES

[1] W. Zhao, R. Chellappa, J. Phillips, A. Rosenfeld, "Face Recognition in Still and Video Images: A Literature Surrey", *ACM Computing Surveys*, Vol. 35, Dec. 2003, pp. 399-458

[2] M. Turk, "A Random Walk through Eigenspace", *IEICE Trans. Inf. & Syst.*, Vol. E84-D, No. 12, December 2001, pp. 1586-1595

[3] M. Turk, A. Pentland, "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, 1991, pp. 71-86

[4] M. Turk, A. Pentland, "Face Recognition using Eigenfaces", *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, June 1991, pp. 586-591

[5] M.S. Bartlett, J.R. Movellan, T.J. Sejnowski, "Face Recognition by Independent Component Analysis", *IEEE Trans. on Neural Networks*, Vol. 13, No. 6, November 2002, pp. 1450-1464

[6] P. Belhumeur, J. Hespanha, D. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", *Proc. of the Fourth European Conference on Computer Vision*, Vol. 1, 14-18 April 1996, Cambridge, UK, pp. 45-58

[7] A. Martinez, A. Kak, "PCA versus LDA", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 2, February 2001, pp. 228-233

[8] P.J. Phillips, H. Moon, S.A. Rizvi, P.J. Rauss, "The FERET Evaluation Methodology for Face Recognition Algorithms", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 10, October 2000, pp. 1090-1104

[9] Yale University Face Image Database, http://cvc.yale.edu/projects/yalefaces/yalefaces.html

[10] AR Face Database, http://rvl1.ecn.purdue.edu/aleix/aleix_face_DB.html

[11] XM2VTS Face Database, http://www.ee.surrey.ac.uk/Research/VSSP/xm2vtsdb

[12] CMU PIE Face Database, http://www.ri.cmu.edu/projects/project_418.html

[13] J.F. Cardoso, "Infomax and Maximum Likelihood for Source Separation", *IEEE Letters on Signal Processing*, Vol. 4, No. 4, April 1997, pp. 112-114

[14] B. Draper, K. Baek, M.S. Bartlett, J.R. Beveridge, "Recognizing Faces with PCA and ICA", *Computer Vision and Image Understanding (Special Issue on Face Recognition)*, Vol. 91, Issues 1-2, July-August 2003, pp. 115-137

[15] M. Zibulevsky and B. A. Pearlmutter, "Blind Separation of Sources with Sparse Representations in a Given Signal Dictionary", *International Workshop on Independent Component Analysis and Blind Signal Separation*, 19-22 June 2000, Helsinki, Finland

[16] A. Hyvarinen, "The Fixed-point Algorithm and Maximum Likelihood Estimation for Independent Component Analysis", *Neural Processing Letters*, Vol. 10, Issue 1, August 1999, pp. 1-5

[17] P. Navarrete, J. Ruiz-del-Solar, "Analysis and Comparison of Eigenspace-Based Face Recognition Approaches", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 16, No. 7, November 2002, pp. 817-830

[18] W. Zhao, R. Chellappa, A. Krishnaswamy, "Discriminant Analysis of Principal Components for Face Recognition", *Proc. of the 3rd IEEE International Conference on Automatic Face and Gesture Recognition*, 14-16 April 1998, Nara, Japan, pp. 336-341

[19] J.R. Beveridge, K. She, B. Draper, G.H. Givens, "A Nonparametric Statistical Comparison of Principal Component and Linear Discriminant Subspaces for Face Recognition", *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, December 2001, Kaui, HI, USA, pp. 535- 542

[20] K. Baek, B. Draper, J.R. Beveridge, K. She, "PCA vs. ICA: A Comparison on the FERET Data Set", *Proc. of the Fourth International Conference on Computer Vision, Pattern Recognition and Image Processing*, Durham, NC, USA, 8-14 March 2002, pp. 824-827

[21] C. Liu, H. Wechsler, "Comparative Assessment of Independent Component Analysis (ICA) for Face Recognition", *Second International Conference on Audio- and Video-based Biometric Person Authentication*, Washington D.C., USA, 22-23 March 1999