Eigenspace-Based Face Recognition: A Comparative Study of Different Approaches

Javier Ruiz-del-Solar, *Member, IEEE,* and Pablo Navarrete, *Student Member, IEEE*

Abstract—**Eigenspace-based face recognition corresponds to one of the most successful methodologies for the computational recognition of faces in digital images. Starting with the Eigenface-Algorithm, different eigenspace-based approaches for the recognition of faces have been proposed. They differ mostly in the kind of projection method used (standard, differential, or kernel eigenspace), in the projection algorithm employed, in the use of simple or differential images before/after projection, and in the similarity matching criterion or classification method employed. The aim of this paper is to present an independent comparative study among some of the main eigenspace-based approaches. We believe that carrying out independent studies is relevant, since comparisons are normally performed using the implementations of the research groups that have proposed each method, which does not consider completely equal working conditions for the algorithms. Very often, a contest between the abilities of the research groups rather than a comparison between methods is performed. This study considers theoretical aspects as well as simulations performed using the Yale Face Database, a database with few classes and several images per class, and FERET, a database with many classes and few images per class.**

Index Terms—**Eigenfaces, eigenspace-based analysis, face recognition, face recognition algorithms' comparisons.**

I. INTRODUCTION

FACE RECOGNITION is a high-dimensional pattern recognition problem. Even low-resolution face images generate huge dimensional feature spaces (20 000 dimensions in the case of a 100×200 pixels face image). In addition to the problems of large computational complexity and memory storage, this high dimensionality makes very difficult to obtain statistical models of the input space using well-defined parametric models. Moreover, this last aspect is further stressed given the fact that only few samples for each class (1–3) are normally available for the system training. However, the intrinsic dimensionality of the face space is much lower than the dimensionality of the image space, since faces are similar in appearance and contain significant statistical regularities. This fact is the starting point of the use of eigenspace-based methods for reducing the dimensionality of the input face space. Standard as well as differential and kernel eigenspace approaches have been presented in the literature to overcome the aforementioned problems. Given that similar troubles are normally found in many biometric applications, we believe that some of the eigenspace-based methods outlined

and compared in this work can be applied in the implementation of other biometric systems (signature, fingerprint, iris, etc.).

Eigenspace-based methods, mostly derived from the Eigenface-Algorithm [19], project input faces onto a dimensionally reduced space where the recognition is carried out, performing a holistic analysis of the faces. Different eigenspace-based methods have been proposed. They differ mostly in the kind of projection/decomposition approach used (standard, differential or kernel eigenspace), in the projection algorithm employed, in the use of simple or differential images before/after projection, and in the similarity matching criterion or classification method employed. The aim of this paper is to present an independent comparative study among some of these different approaches. We believe that carrying out independent studies is relevant, because comparisons are normally performed using the own implementations of the research groups that have proposed each method (e.g., in FERET contests), which does not consider completely equal working conditions (e.g., exactly the same preprocessing steps). Very often, a contest between the abilities of the research groups rather than a comparison between methods is performed. Additionally, not all the possible implementations are considered (e.g., p projection methods with q similarity criteria), but only the ones that some groups have decided to use.

This study corresponds to an extension of the one presented in [12]. It considers standard, differential, and kernel eigenspace methods. In the case of the standard methods, three different projection algorithms [Principal Component Analysis (PCA), Fisher Linear Discriminant (FLD), and Evolutionary Pursuit (EP)], and five similarity matching criteria [Euclidean-, Cosine-, and Mahalanobis-distance; SOM-Clustering; and Fuzzy Feature Contrast (FFC)], were considered. In the case of differential eigenspace methods, two approaches were used: predifferential [13] and postdifferential [15]. In both cases, two classification methods [Bayesian, and Support Vector Machine (SVM classification)], were employed. Finally, regarding kernel eigenspace methods [9], Kernel PCA (KPCA) and Kernel Fisher Discriminant (KFD) were used together with the five similarity measures employed in the standard eigenspace methods.

This comparative study considers theoretical aspects as well as simulations performed using the Yale Face Database, a database with few classes and several images per class, and FERET, a database with many classes and few images per class. It is important to use both kinds of databases for performing such a study, because, as will be shown in this work, some properties of the methods, for example their generalization ability, change depending on the number of classes taken under consideration.

Preprocessing aspects such as face alignment, masking, illumination compensation and so on, are kept unchanged in all the approaches and their different implementations.

This paper is structured as follows. In Section II, different approaches for the eigenspace-based recognition of faces are described. In Section III, a comparative study among these approaches is presented. Finally, some conclusions regarding this work are given in Section IV.

II. EIGENSPACE-BASED RECOGNITION OF FACES

Standard eigenspace-based approaches project input faces onto a dimensionally reduced space where the recognition is carried out. In 1987, Sirovich and Kirby used PCA in order to obtain a reduced representation of face images [17]. Then, in 1991, Turk and Pentland used PCA projections as the feature vectors to solve the problem of face recognition using the Euclidean distance as similarity function [19]. This system, later called *Eigenfaces*, was the first eigenspace-based face recognition approach and, from then on, many eigenspace-based systems have been proposed using different projection methods and similarity functions. In particular, Belhumeur *et al.* proposed in 1997 the use of FLD as projection algorithm in the so-called *Fisherfaces* system [1]. In all standard eigenspace-based approaches a similarity function, which works as a nearest-neighbor classifier [3], is employed.

In 1997, Pentland and Moghaddam proposed a differential eigenspace-based approach that allows the application of statistical analysis in the recognition process [13]. The main idea is to work with differences between face images, rather than with single face images. In this way the recognition problem becomes a two-class problem, because the so-called "differential image" contains information of whether the two subtracted images belong to the same class or to different classes. In this case the number of training images per class increases so that statistical information becomes available, and a statistical classifier can be used for performing the recognition. The system proposed in [13] used Dual-PCA projections and a Bayesian classifier. Following the same approach, a system using Single-PCA projections and a SVM classifier was proposed in [15].

In the differential approach all the face images need to be stored in the database, which slows down the recognition process. This is a serious drawback in practical implementations. To overcome this drawback a so-called post-differential approach was proposed in [15]. Under this approach, differences between reduced face vectors instead of differences between face images are used. This allows a decrease in the number of computations and the required storage capacity (only reduced face vectors are stored in the database), without losing the recognition performance of the differential approaches. Both Bayesian and SVM classifiers were used to implement this approach in [15].

On the other hand, kernel eigenspace methods were proposed for increasing the generalization ability of eigenspace methods by increasing the dimensionality of the input space instead of using differences of face images or reduced vectors. In this case KPCA and KFD, which are nonlinear extensions of PCA and FLD, respectively, are used as projection algorithms. The main idea behind these projection algorithms is to use linear methods applied to high-dimensional mapped vectors instead of the original vectors, and at the same time to avoid the explicit mapping of these vectors by means of the so-called "kernel-trick" (the same strategy is employed in SVM). As in the case of the standard eigenspace methods, a similarity function, which works as a nearest-neighbor classifier, is employed. A kernel-based system for the recognition of faces was proposed in [10]. This system uses either KPCA or KFD as the projection algorithm.

Standard, differential, and kernel eigenspace approaches for the recognition of faces are described in the following subsections.

A. Standard Eigenspace Face Recognition

Fig. 1 shows the block diagram of a generic standard eigenspace-based face recognition system. Standard eigenspace-based approaches approximate the face vectors (face images) by lower dimensional feature vectors. These approaches consider an off-line phase or training, where the *projection matrix* ($\mathbf{W} \in R^{N \times m}$), the one that achieves the dimensional reduction, is obtained using all the database face images. In the off-line phase, the *mean face* ($\bar{x}$) and the reduced representation of each database image ($\mathbf{p}^k$) are also calculated. The recognition process works as follows. A preprocessing module transforms the face image into a unitary vector (normalization module in the case of Fig. 1) and then performs a subtraction of the mean face. The resulting vector is projected using the projection matrix that depends on the eigenspace method being used (PCA, FLD, etc.). This projection corresponds to a dimensional reduction of the input, starting with vectors in R^N (where N is the dimension of the image vectors) and obtaining projected vectors $\mathbf{q}$ in R^m, with $m < N$ (usually $\mathrm{m} \ll \mathrm{N}$). Then, the similarity of $\mathbf{q}$ with each of the reduced vectors $\mathbf{p}^k (\mathbf{p}^k \in R^m)$ is computed using a certain criterion of similarity (Euclidean distance for example). The class of the most similar vector is the result of the recognition process; i.e., the identity of the face. In addition, a rejection system for unknown faces is used if the similarity matching measure is not good enough.

1) Projection Methods: The projection methods employed in this work for the reduction of dimensionality are PCA [19], FLD [4], and EP [5]. PCA is a general method for identifying the linear directions in which a set of vectors are best represented in a least-squares sense, allowing a dimensional reduction by choosing the directions of largest variance. The theoretical solution of this problem is well known and is obtained by solving the eigensystem of the correlation matrix $\mathbf{R}$. On the other hand, FLD searches for the projection axes on which the input vectors of different classes are far away from each other (similar to PCA), and at the same time input vectors of a same class are close to each other. The solution of this problem is obtained by solving the general eigensystem for the so-called within-class and between-class scatter matrices. Computational aspects of the PCA and FLD algorithm implementation can be found in [12]. The main difference between PCA and FLD is that PCA seeks a projection that best represents data in a least-squares sense (good
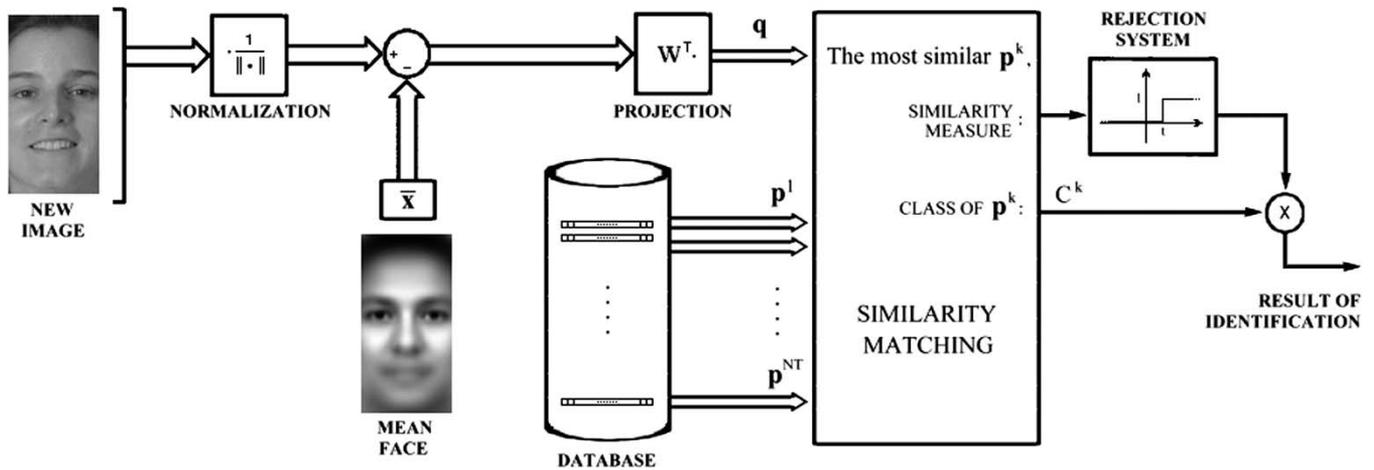
Fig. 1.   Block diagram of a generic standard eigenspace-based face recognition system.

representation), and FLD seeks a projection that best separates the data in a least-squares sense (good discrimination) [3].

The advantage of FLD against PCA is that the information kept in the dimensional reduction is better for recognition purposes. However, there are some drawbacks. For example, FLD uses the particular class information, so it is recommended to have many images per class in the training process, or at least a good characterization of each class. In other words, in PCA the convergence of the **R** estimator depends mostly on the total number of target images, but in FLD the convergence of the scatter matrix estimators depends also on the numbers of target images per class. The main drawback of FLD is that it could be over-adjusted on the target images, and then the recognition system may have lack of good generalization that may be reflected in the resulting system's recognition rate.

EP corresponds to an adaptive eigenspace strategy that tries to overcome the drawbacks of PCA and FLD. EP searches adaptively for the best set of projection axes in order to maximize a fitness function, measuring at the same time the classification accuracy and generalization ability of the system. Because the dimension of the solution-space of this problem is too large, it is solved using a specific kind of genetic algorithm called Evolutionary Pursuit (EP) [5]. EP requires an initial set of axes, which are obtained using PCA.

2) Similarity Matching Methods: The main objective of a similarity measure is to define a value that allows the comparison of feature vectors (reduced vectors in eigenspace frameworks). With this measure the identification of a new feature vector will be possible by searching the most similar vector into the database. This is the well-known nearest-neighbor method. One way to define similarity is to use a measure of distance, $d(\mathrm{x}, \mathrm{y})$, in which the similarity between vectors, $S(\mathrm{x}, \mathrm{y})$, is inverse to the distance measure. Another form is to use a correlation measure between vectors. In this work we have used two well-known distance measures, Euclidean and Mahalanobis distance; two correlation-like measures, Cosine distance and SOM Clustering; and one fuzzy measure, FFC [16].

It can be proved that in the PCA space the Mahalanobis distance is equivalent to the Euclidean distance, weighting each

component by the inverse of the correspondent eigenvalue (see demonstration in [11]), and it is often called Whitening (PCA) transformation. The same transformation can be applied to other similarity measures. Taking into account this property, we finally used eight similarity measures in our comparative study: Euclidean, Whitening Euclidean (same as Mahalanobis), Cosine, and Whitening Cosine distances, SOM, and Whitening SOM Clustering, FFC, and Whitening FFC.

Given that the EP projection algorithm already includes a Whitening stage, in this case only four similarity measures (Euclidean, Cosine, SOM Clustering, and FFC) were employed.

B. Differential Eigenspace Face Recognition

As mentioned, differential eigenspace-methods work with differences between vectors, rather than with vectors. In the pre-differential case the vectors to be subtracted correspond to the face images, while in the post-differential case they correspond to reduced face vectors.

1) Predifferential Eigenspace Methods: Fig. 2 shows the block diagram of a generic predifferential eigenspace-based face recognition system. In the off-line phase the projection matrix is computed and in the database the whole face images are stored (NT images). Previously, these face images are centered and scaled so that they are correctly aligned. The recognition process works as follows. The input face image is preprocessed and subtracted from each database image. The result of each subtraction is called "differential image" Δ in R^N. The "differential image" is the key for identification because it contains information of whether the two subtracted images are from the same class or from different classes. In this way the original problem of NC-class becomes a two-class problem. The so-called differential images (NT images) are projected into a reduced space using a certain projection method. Thus, each image is transformed into a reduced differential vector δ in R^m. Thereafter the classification of the reduced differential vectors is performed using any statistical classifier. The result of each classification (S_i) is negative if the subtracted images (each δ) belong to different classes, and positive otherwise. In order to determine the
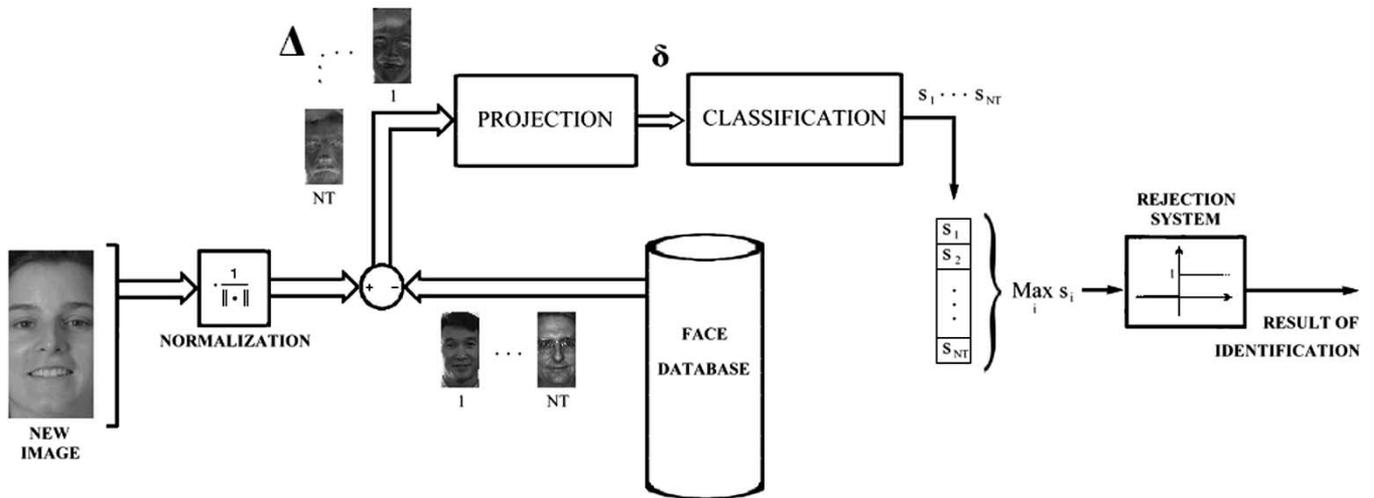
Fig. 2.   Block diagram of a generic pre-differential eigenspace face recognition system.
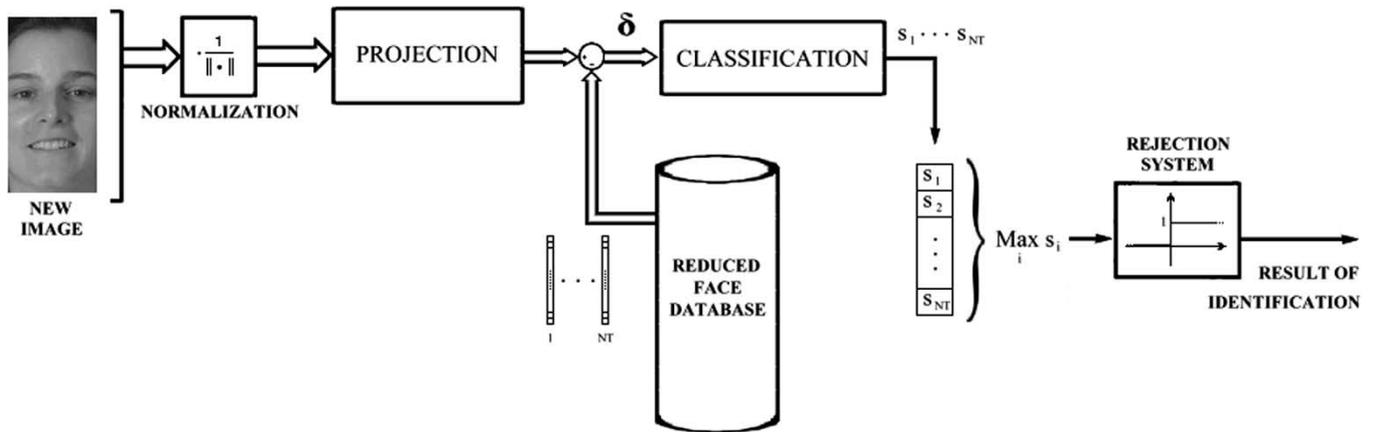

Fig. 3.   Block diagram of a generic postdifferential eigenspace face recognition system.

class of the input face image, the reduced vector with maximum positive classification value is chosen, and the class of its corresponding database image is given as the result of identification. The rejection system acts just when the maximum classification value is negative; i.e., it corresponds to the subtraction of different classes.

Dual-PCA and Single-PCA projections have been used as projection methods. Dual-PCA projections employ two projection matrices: $\mathbf{W}_I \in R^{N \times m_I}$ for intra-classes Ω_I (subtractions within equal classes), and $\mathbf{W}_E \in R^{N \times m_E}$ for extra-classes Ω_E (subtractions between different classes). Single-PCA projections employ a single projection matrix $\mathbf{W} \in R^{N \times m}$ (like normal PCA). In this work we used Dual-PCA projections together with a Bayesian classifier [3], and Single-PCA projections together with a SVM classifier [2].

2) Post-Differential Eigenspace Methods: Fig. 3 shows the block diagram of a generic post-differential eigenspace-based face recognition system. In this approach only the reduced face images (NT vectors) are stored in the database. In the recognition process an input face image is preprocessed and then projected into a reduced space using a certain projection method.

Thereafter, the new reduced face image is subtracted from each database reduced face image. The result of each subtraction is called "post-differential image" δ in R^m. This vector contains information about whether the two subtracted vectors are from the same class or from different classes (intra-classes or extra-classes), and then it works in the same way as a "differential image" projected on the reduced space. The classification module performs the classification of the differential vectors using any statistical classifier. The class of the reduced database vector that has the maximum positive classification value gives the class of the input face image. If the projection module does not significantly change the topology of the differential-image space, then the pre-differential and post-differential approaches should have very similar recognition rates. The rejection system acts just when the maximum classification value is negative; i.e., it corresponds to the subtraction of different classes.

In this work, two different systems were implemented following this approach. The first system uses Single-PCA projections together with a Bayesian classifier, and the second employs Single-PCA projections together with a SVM classifier.
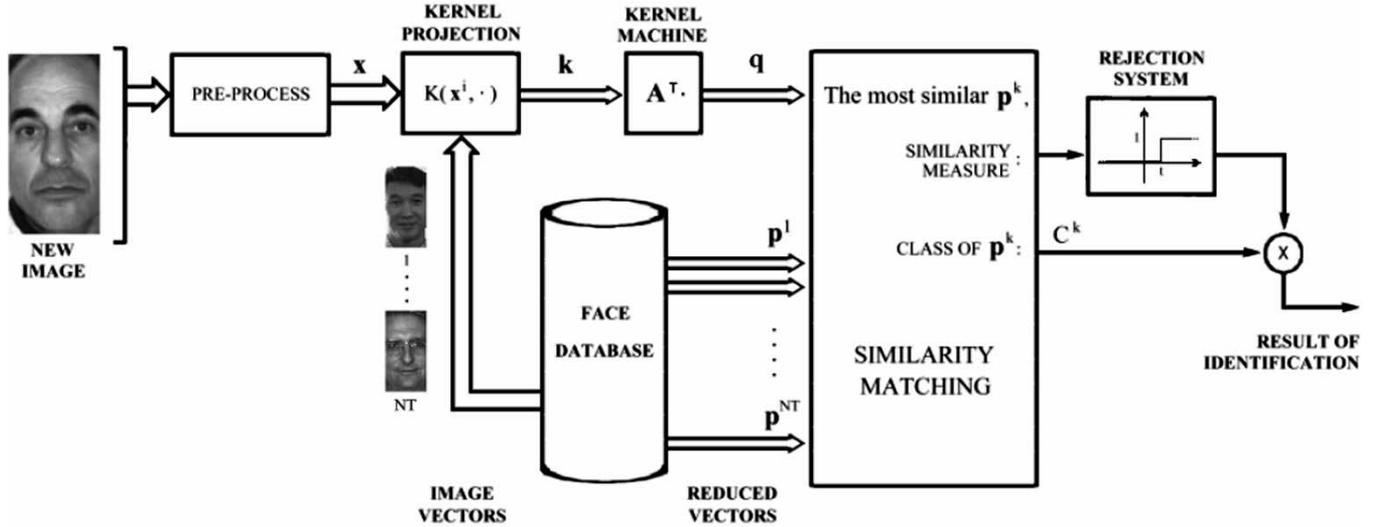
Fig. 4. Block diagram of a generic kernel-based face recognition system.

C. Kernel Eigenspace Face Recognition

Kernel eigenspace algorithms correspond to nonlinear generalizations of linear eigenspace algorithms. This generalization is carried out using kernels and works as follows. If the algorithm to be generalized uses the training vectors only in the form of Euclidean dot products, then it can be "kernelized," and all the dot products such as $\mathbf{x}^T\mathbf{y}$ are replaced by a so-called kernel function $K(\mathbf{x},\mathbf{y})$. If $K(\mathbf{x},\mathbf{y})$ fulfills Mercer's condition; i.e., the operator K is semi-positive definite, then the kernel can be expanded into a series $K(\mathbf{x},\mathbf{y}) = \sum_i \phi_i(\mathbf{x})\phi_i(\mathbf{y})$ [10]. In this way the kernel represents the Euclidean dot product on a different space, called feature space F, on which the original vectors are mapped using the eigenfunctions $\phi : \Re^N \to F$. Depending on the kernel function used, the feature space F can even be of infinite dimension, as it is in the case of Radial Basis Function (RBF) kernel, but the algorithm never works directly in such space.

Kernel eigenspace face recognition systems employ a kernel eigenspace algorithm such as KPCA or KFD for projecting the input face images. They work in the same way as standard-eigenspace methods, but use a nonlinear projection, which is implemented in two steps. Fig. 4 shows the block diagram of a generic kernel-based face recognition system.

Under the kernel approach the projection of the input face images should be carried out in two steps: *kernel projection* and *kernel machine projection* [9]. In the kernel projection step, each preprocessed input face image $\mathbf{x} \in \Re^N$ (mapped onto the feature space $\phi(\mathbf{x}) \in F$) is projected on the NT support images (mapped in the same feature space), using the kernel function $K : \Re^N \times \Re^N \to \Re^{NT}$ (see computational details in [9]). To accomplish that, the NT face images need to be stored in the face database. This is due to the fact that kernel machines need all the image vectors in order to reproduce the eigenvectors in the feature space F [10]. In the kernel machine projection step, the parameters of the given kernel machine $\mathbf{A}^T \in M^{NT \times m}$ (see computational details in [9]) are applied to the kernel projection

vector $\mathrm{k} \in \Re^{NT}$ in order to obtain the feature vector $\mathbf{q} \in \Re^m$. After projection, the similarity of the feature vector $\mathbf{q}$ with the reduced vectors $\mathbf{p}^k, \mathbf{p}^k \in \Re^m$ is computed using a certain criterion of similarity. The class of the most similar vector is the result of the recognition process; i.e., the identity of the face. In addition, a *Rejection System* for unknown faces is used if the similarity matching measure is not good enough.

In the off-line phase the matrices $K()$ and $\mathbf{A}^T$ are computed, and using these matrices the reduced vectors $\mathbf{p}^k$ are calculated by projecting the face images. The main drawback of this approach is that all NT face images and all NT reduced vectors should be stored in the database.

In this work we have used KPCA and KFD as projection algorithms. As similarity measures we used the same 10 similarity measures used for the standard eigenspace approaches (see Section II-A1).

III. COMPARATIVE STUDY

The comparative study presented in this section considers theoretical aspects as well as simulations performed using the Yale Face Database, a database with few classes and several images per class, and FERET, a database with many classes and few images per class. It is important to use both kinds of databases, because some properties of the methods, for example their generalization ability, change depending on the number of classes under consideration. Pre-processing aspects in all the approaches are kept unchanged.

A. Preprocessing

The preprocessing stage used in this work includes the following sequential operations.

- *Face alignment* by centering the eyes in the same relative positions.
- *Window resizing* by scaling input face images using fixed proportions to obtain face images of 100×200 pixels.

TABLE I
STANDARD EIGENSPACE (YALE DATABASE)

projection method	images per class	axes	Euclidean	cos()	SOM	FFC	whitening Euclidean	whitening cos()	whitening SOM	whitening FFC
PCA		49	95,7 2,7	95,8 2,7	94,2 2,8	81,8 5,4	83,3 5,9	89,3 4,1	88,8 3,8	81,8 5,4
FLD	6	14	94,6 2,1	95,2 2,5	93,5 2,4	85,9 5,2	**97,2** **2,2**	97,0 2,5	96,7 3,5	90,8 5,5
EP		21	89,8 4,1	94,3 4,0	92,7 4,2	85,2 3,8	-	-	-	-
PCA		42	94,0 2,5	94,1 2,5	92,5 3,1	76,8 10,4	82,2 7,2	87,7 5,6	87,3 5,8	76,8 10,4
FLD	5	14	94,0 3,2	94,3 2,6	92,8 2,8	87,3 5,8	**95,0** **3,9**	94,2 4,7	93,7 4,5	87,7 6,1
EP		16	93,7 3,2	93,9 2,6	92,1 3,3	88,2 3,6	-	-	-	-
PCA		35	93,4 1,9	93,4 2,1	91,6 2,3	78,7 5,5	85,4 4,0	88,0 4,0	79,2 5,3	78,7 5,5
FLD	4	14	92,9 2,4	93,5 2,4	93,8 2,5	84,7 3,8	**94,4** **2,1**	92,9 3,9	93,1 4,2	85,1 5,7
EP		14	92,3 2,6	92,9 2,5	91,8 2,6	85,3	-	-	-	-
PCA		28	91,9 2,5	92,4 2,2	88,5 2,7	78,6 6,8	84,2 4,0	86,0 4,5	84,0 5,6	78,6 6,8
FLD	3	14	89,8 4,5	90,9 4,4	85,7 4,6	81,6 5,5	**93,0** **2,2**	92,0 2,8	92,1 2,8	83,9 5,0
EP		14	84,2 3,5	91,2 4,4	88,5 4,4	82,1 5,1	-	-	-	-
PCA		21	88,9 5,0	88,9 5,0	79,1 6,1	75,5 6,9	83,4 6,3	85,1 4,0	75,2 6,6	75,5 6,9
FLD	2	14	88,5 3,4	88,1 4,2	77,2 4,2	79,6 6,1	**89,9** **4,1**	88,1 2,8	86,6 3,4	77,2 7,2
EP		14	77,9 4,8	88,4 5,1	78,5 3,9	78,4 5,3	-	-	-	-

TABLE II
PRE-DIFFERENTIAL EIGENSPACE. (i)/(e) INDICATES INTRA/EXTRACLASSES
(YALE DATABASE)

images per class	axes Dual-PCA Single-PCA		Dual-PCA +Bayesian	Single-PCA +SVM
6	168 (i) / 182 (e)	334	**97,5** 3,6	96,8 3,4
5	115 (i) / 126 (e)	153	**95,5** 3,7	94,6 4,1
4	85 (i) / 84 (e)	121	**93,2** 3,0	92,1 3,4
3	39 (i) / 41 (e)	74	90,4 3,2	**91,8** **4,4**
2	12 (i) / 14 (e)	23	89,5 3,7	**90,3** **4,6**

TABLE III
POSTDIFFERENTIAL EIGENSPACE (YALE DATABASE)

images per class	axes	Dual-PCA +Bayesian	Single-PCA +SVM
6	342	95,6 4,3	**97,4** **3,6**
5	151	92,1 5,3	**93,8** **4,5**
4	115	92,3 4,2	**92,6** **3,0**
3	81	87,9 3,7	**91,5** **3,8**
2	22	86,7 4,8	**88,9** **5,1**

- *Image Masking* by cutting the image corners to avoid processing border pixels not corresponding to the image face.
- *Illumination gradient compensation* by subtracting a best-fit brightness plane to each image. This method allows to compensate heavy shadows caused by extreme lighting angles.
- *Histogram equalization* by spreading the energy of all intensity pixel values in the image.
- *Normalization* to make all input image vectors of the same energy.

B. Simulations Using the Yale Face Image Database

The first simulations were carried out using the Yale University Face Image Database [20]. We employed 150 images of 15 different classes (10 images per class). In Table I, we show the results of several simulations for the standard eigenspace approaches. For each simulation we used a fixed number of training images, using the same type of images per class, according with the Yale database specification. In order to obtain representative results we take the average of 20 different sets of images for each fixed number of training images. All the images not used for training were used for testing. In Tables II and III,

TABLE IV
KERNEL EIGENSPACE (YALE DATABASE)

projection method	images per class	axes	Euclidean	cos()	SOM	FFC	whitening Euclidean	whitening cos()	whitening SOM	whitening FFC
KPCA	6	89	96,1 2,7	96,1 2,7	95,1 2,6	82,7 8,9	92,6 4,6	90,7 5,9	88,3 6,3	82,7 8,9
KFD		14	**96,9** 2,2	96,8 1,9	95,1 1,9	92,4 4,2	96,3 2,6	93,9 3,6	94,5 4,2	89,8 6,0
KPCA	5	74	94,5 2,5	94,5 2,5	92,3 2,6	82,9 9,9	88,9 7,5	87,7 7,6	87,9 9,2	82,9 9,9
KFD		14	94,9 4,1	**95,4** 2,8	91,4 2,9	89,4 5,2	94,5 4,1	92,3 5,4	91,8 5,6	87,6 6,1
KPCA	4	59	93,7 1,9	93,7 1,9	90,6 2,2	84,6 6,2	89,9 4,9	88,1 4,6	83,4 5,8	84,6 6,2
KFD		14	94,1 3,4	**95,7** 2,4	91,5 3,4	89,1 4,3	92,9 2,6	91,5 3,7	91,5 4,1	84,3 4,9
KPCA	3	44	92,5 1,9	92,5 1,9	90,5 2,5	82,6 5,5	90,3 3,3	88,1 3,6	83,8 5,4	82,6 5,5
KFD		14	92,7 2,6	**94,0** 1,8	90,3 2,0	87,1 5,2	93,0 2,4	91,3 3,0	90,2 3,3	82,3 4,8
KPCA	2	29	89,9 4,3	89,9 4,3	85,2 4,4	76,2 7,8	90,2 3,7	87,7 3,2	83,3 7,5	76,2 7,8
KFD		14	90,4 2,6	**92,3** 3,6	88,4 3,6	82,0 4,7	89,1 4,1	87,3 4,1	84,5 5,4	77,5 5,7

TABLE V
STANDARD EIGENSPACE (FERET DATABASE, 127 CLASSES)

projection method	images per class	axes	Euclidean	cos( )	FFC	whitening Euclidean	whitening cos( )	whitening FFC
PCA		192	95.3 2.2	**95.7** 1.7	90.9 2.8	81.9 4.5	93.3 0.6	90.9 2.8
FLD	3	126	95.3 3.3	95.3 2.2	92.9 1.1	86.6 12.2	89.0 8.9	90.6 5.6
EP		109	94.8 2.4	94.8 2.9	90.3 0.8	–	–	–
PCA		149	86.8 2.5	87.0 2.8	84.1 3.1	78.0 2.2	86.6 1.1	84.1 3.1
FLD	2	126	88.2 3.9	87.6 4.2	86.6 3.9	81.9 3.3	**88.8** 3.1	87.4 2.8
EP		107	86.9 3.0	87.9 3.2	85.7 2.1	–	–	–

TABLE VI
PRE-DIFFERENTIAL EIGENSPACE. (i)/(e) INDICATES INTRA/EXTRACLASSES (FERET DATABASE, 127 CLASSES)

images per class	axes		Dual-PCA+ Bayesian	Single-PCA+ SVM
	Dual-PCA	Single-PCA		
3	122 (i) / 124 (e)	224	93.1 0.9	**95.5** 1.4
2	95 (i) / 103 (e)	125	86.6 0.5	**86.7** 1.3

we show the results of several simulations using pre-differential and post-differential approaches. We used equal *a priori* probabilities for the Bayes-based methods, $P(\Omega_{\mathrm{I}}) = P(\Omega_{\mathrm{E}})$, and a penalty for nonseparable cases, $C = 0.01$, in the SVM classification method (see details in [15]). In Table IV, we show the results of several simulations for the kernel approaches. The selected kernel function was a RBF with $\sigma = 0.5$. In KFD the regularization parameter was $\mu = 0.05$ (see details in [10]). In Tables I–IV, the best results obtained in each experiment, for each approach (standard, pre- and post-differential, and kernel), are indicated in bold.

C. Simulations Using FERET

In order to test the described approaches using a large database, we made simulations using the FERET database [14]. We used a target set with 762 images of 254 different classes (3 images per class), and a query set of 254 images (1 image per class). Using these sets we carried out two different simulations. In the first simulation (Tables V–VIII), we split the target and query sets in two disjoint sets, and we performed the experiments considering 127 classes. In the second simulation (Tables IX–XII), we employed the full target and query sets, and we carried out the experiments considering the 254 classes. In

TABLE VII
POST-DIFFERENTIAL EIGENSPACE (FERET DATABASE, 127 CLASSES)

images per class	axes	Dual-PCA + Bayesian	Single-PCA + SVM
3	126	93.4 1.3	**94.3** 0.5
2	211	87.6 2.1	**88.6** 1.8

Tables V and IX we show the results of simulations for the standard approach. In these tables the SOM-based clustering was not included since in these tests the number of classes (127 or 254) is much larger than the number of images per class (3), and the SOM training process (self-organization) is very difficult. In Tables VI, VII, X, and XI, we show the results of simulations using pre-differential and post-differential approaches. The same parameters as before were used for the Bayesian and SVM classifiers (see Section III-B). In Tables VIII and XII, we show the results of simulations for the kernel approaches. The selected kernel function was a RBF with $\sigma = 0.5$. In KFD, the regularization parameter was $\mu = 0.05$ (see details in [10]). In Tables V–XII, the best results obtained in each experiment, for each approach (standard, pre- and post-differential and kernel), are indicated in bold.

Tables I, II, III, and IV: Mean recognition rates using the Yale database (15 classes) and different numbers of training images per class, and taking the average of 20 different training sets. The small numbers are standard deviations. All results consider the top 1 match. Whitening is equivalent to use a Mahalanobis distance in a projection space [9]. The EP projection algorithm already includes a whitening stage.

Tables V, VI, VII, and VIII: Mean recognition rates for standard approaches using FERET, 127 different classes, and taking the average of two different training sets. The small numbers are standard deviations. All results consider the top 1 match for recognition. Whitening is equivalent to use a Mahalanobis distance in a projection space [9]. The EP projection algorithm already includes a whitening stage.

Tables IX, X, XI, and XII: Mean recognition rates for standard approaches using FERET and 254 different classes. All results consider the top 1 match for recognition. Whitening is equivalent to use a Mahalanobis distance in a projection space [9]. The EP projection algorithm already includes a whitening stage.

D. Analysis of Results

By analyzing the Yale database simulations (Tables I–IV), the following can be concluded.

– When using the standard approach (Table I), the best results are obtained with the FLD-Whitening-Euclidean combination. Using other FLD combinations very similar results are obtained (consider the standard deviation information). These results confirm the better theoretical discrimination ability of FLD over PCA, and the results reported when Fisherfaces was proposed for the first time [1]. It should be remembered that the FLD algorithm

obtains projection axes that best separates the input data in a least-squares sense (see 2.1.1). Practical problems of FLD, for example the "small sample size" problem [7] produced when the number of target images per class is small, cannot be observed in these experiments.

– Considering the differential approaches (Tables II and III), the results obtained using the predifferential approach are slightly better than the ones obtained with the post-differential approach. When the number of images per classes is low (2 or 3) the results for both approaches are very similar (consider the standard deviation information). The Bayesian classifier and SVM give similar results in both cases.

– Regarding the kernel approaches (Table IV), KFD gives better results than KPCA using fewer projection axes. As in the case of FLD, the reason seems to be the better theoretical discrimination ability of KFD over KPCA. In both cases very similar results are obtained using Euclidian or cosines distances.

– The Yale database contains few classes (15) and several images per class (2–6), and in general the best results obtained with each approach are very similar. Differences are seen only when the number of images per classes is low (2 or 3). In this last case, the approaches with better generalization ability (the kernel ones) obtain better results. By looking at the standard deviation information it can also be noted that the kernel approaches have a smaller variability in their results. Regarding the number of projection axes employed, FLD and KFD use less axes (14). This number corresponds to the number of classes (15 in this case)—1 (see demonstration in [3]).

As previously explained, two different simulations where performed with the FERET-database, one using 127 classes and two disjoint training and two disjoint target sets (Tables V–VIII), and another using 254 classes and just one training and one target set (Tables IX–XII). By analyzing these simulations, the following can be concluded.

– When using the standard approach and 127 classes (Table V) the results obtained with PCA and FLD are very similar (consider the standard deviation information). However, when 254 classes are employed (Table IX) the best results are obtained with PCA. The reason seems to be the better generalization ability of PCA over FLD. As mentioned in Section II-A2, while in PCA the convergence of the correlation matrix estimator depends on the total number of samples, in KFD the convergence of the scatter matrices estimators depends on the number of target images per class. These results are in accordance with the ones obtained in [8], which concludes that PCA might outperform LDA when the number of samples per class is small or when the training data nonuniformly sample the underlying distribution, which seems to be the case in these experiments.

– When analyzing the number of axes employed in the standard methods (Tables V and IX), it can be seen that EP employs a lower numbers of axes than PCA and FLD. Considering that the results obtained with EP are just

TABLE VIII
KERNEL EIGENSPACE (FERET DATABASE, 127 CLASSES)

projection method	images per class	axes	Euclidean	cos( )	FFC	whitening Euclidean	whitening cos( )	whitening FFC
KPCA	3	380	95.7 1.7	95.7 1.7	91.7 1.7	90.6 5.6	95.7 0.6	91.7 1.7
KFD		126	**96.5** 1.7	**96.5** 1.7	94.9 2.8	89.8 3.3	83.9 3.9	86.6 1.1
KPCA	2	253	88.0 3.1	88.0 3.1	86.4 1.9	90.0 3.6	**90.2** 3.9	86.4 1.9
KFD		126	89.4 2.8	89.0 3.3	89.6 1.4	86.0 0.8	80.5 1.4	83.3 0.3

TABLE IX
STANDARD EIGENSPACE (FERET DATABASE, 255 CLASSES)

projection method	images per class	axes	Euclidean	cos()	FFC	whitening Euclidean	whitening cos()	whitening FFC
PCA		316	**94,1**	**94,1**	87,4	77,6	92,5	87,4
FLD	3	253	92,5	92,1	91,7	79,9	92,9	90,9
EP		218	92,3	91,8	91,5	-	-	-
PCA		252	86,4	**86,8**	81,5	73,2	85,6	81,5
FLD	2	253	85,2	85,0	82,9	73,4	79,9	83,1
EP		194	85,7	86,3	83,7	-	-	-

TABLE X
PRE-DIFFERENTIAL EIGENSPACE. (i)/(e) INDICATES INTRA/EXTRACLASSES
(FERET DATABASE, 255 CLASSES)

images per class	axes Dual-PCA Single-PCA		Dual-PCA +Bayesian	Single-PCA +SVM
3	148 (i) / 156 (e)	247	94,2	**94,8**
2	106 (i) / 128 (e)	192	**89,0**	88,7

TABLE XI
POST-DIFFERENTIAL EIGENSPACE (FERET DATABASE, 255 CLASSES)

images per class	axes	Dual-PCA +Bayesian	Single-PCA +SVM
3	253	92,1	**95,2**
2	199	87,3	**88,5**

slightly lower than those obtained with PCA and FLD, EP obtains the best compromise between a high recognition rate and a low-processing time, which depends on the number of axes.

– In the standard approaches, Euclidian or cosines distances can be used as similarity measures with comparable results. Note that in most of the cases the nonwhitening distances outperform their whitening counterparts.

– Considering the differential approaches (Tables VI, VII, X, and XI), the results obtained using the predifferential and the postdifferential approaches are almost identical.

In both cases the Bayesian classifier and SVM give similar results. When comparing the differential approaches against the standard ones, it can be observed that the results are rather similar when 127 classes were considered. However, when 254 classes were employed, the differential approaches outperform the standard ones. The reason seems to be the better generalization ability of the differential approaches, derived from the fact that more data is available for training, because by using differential images or differential vectors the recognition task becomes a two-class problem.

– Regarding the kernel approaches (Tables VIII and XII), results obtained with KFD and KPCA are very similar. However, KFD gives better results when the number of images per class is 3, and KPCA gives better results when the number of images per class is 2. The reason seems to be the better generalization ability of KPCA over KFD, which is tested when the identification of either 127 or 254 classes is solved using just 2 training images per class. By analyzing the number of employed axes, it can be seen that KFD employs a much lower numbers of axes than KPCA. In this way KFD obtains the best compromise between a high recognition rate and a low processing time.

– The FERET database contains many classes and few images per class (2–3). For this reason, the best results are obtained when using the approaches with better generalization capabilities; i.e., the kernel ones. Kernel approaches achieve better generalization ability by increasing the dimensionality of the input space [10]. Predifferential and postdifferential approaches also obtain good generalization results. However, we believe that further theoretical

TABLE XII
KERNEL EIGENSPACE (FERET DATABASE, 255 CLASSES)

projection method	images per class	axes	Euclidean	cos()	FFC	whitening Euclidean	whitening cos()	whitening FFC
KPCA	3	761	94,5	94,5	85,4	79,5	95,3	85,4
KFD		253	95,3	94,5	**95,7**	82,7	75,2	60,2
KPCA	2	507	86,6	86,6	83,3	**89,6**	89,4	83,3
KFD		253	87,8	87,8	88,6	77,2	71,9	62,0

and practical comparative studies between kernel and differential approaches are required.

Other issues to consider.

– Post-differential approaches are two to five times faster than the pre-differential ones. Taking into account their similar recognition rates, post-differential approaches are the best differential alternative.

– Kernel-projections are 2 to 3 times slower than linear projections due to the use of the support images (all the database images). Another drawback of these methods is that the kernel parameters adjustment is very difficult and data dependant.

– In all approaches the selection of the number of employed axes is a very important issue. Normally, the number of selected axes for each algorithm corresponds to the minimal number of axes required for obtaining an acceptable recognition rate. In the special case of FLD and KFD the employed number of axes is the number of classes—1 [3]. However, there are some quantitative criteria for choosing the right number of axes, for example the RMSE—Residual Mean Square Error [18] for the PCA algorithm, and the RFP—Residual Fisher Parameter [12] for the FLD algorithm. In this last case choosing the number of axes equal to the number of classes—1 gives a very small RFP.

IV. CONCLUSION

The aim of this paper was to present an independent comparative study among different eigenspace-based approaches. The study considered standard, differential and kernel eigenspace methods. In the case of the standard methods, three different projection algorithms (PCA, FLD, and EP) and eight different similarity measures (Euclidean, Whitening Euclidean (Mahalanobis), Cosine Distances, Whitening Cosine Distances, SOM Clustering, Whitening SOM Clustering, FFC, and Whitening FFC) were considered. In the case of the differential methods, two approaches were used: the pre-differential and the post-differential. In both cases Bayesian and SVM classification were employed. Finally, regarding kernel methods, KPCA and KFD were used together with the eight similarity measures employed for the standard approaches.

Simulations were performed using the Yale Face Database, a database with few classes and several images per class, and FERET, a database with many classes and few images per class. By looking at the results, the following can be concluded.

– Considering recognition rates, generalization ability, and processing time, the best results were obtained with the post-differential approach, using either a Bayesian Classifier or SVM.

– In the specific case of the Yale Face Database, where the requirements are not very high, any of the compared approaches gives rather similar results. Thanks to their simplicity, Eigenfaces or Fisherfaces are probably the best alternatives.

– Although kernel methods obtain the best recognition rates, they suffer from problems such as low processing speed and the difficulty to adjust the kernel parameters. The first drawback could be overcome by using a kind of support vector in the KPCA and KFD algorithms, like the ones used in SVM. The problem is that these support vectors do not exist at this time. This is a problem to be tackled by kernel-machine researchers in the next few years.

In future work, we would like to extend our study by considering other kernel approaches and algorithms, for example ICA (Independent Component Analysis), Kernel-ICA, and new algorithms as F-LDA (Fractional-step Linear Discriminant Analysis) [6] and DF-LDA (Direct F-LDA) [7] that improve standard FLD.

REFERENCES

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

[2] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[3] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed.. USA: John Wiley & Sons, 2001.

[4] R. A. Fisher, "The use of multiple measures in taxonomic problems," *Ann. Eugenics*, II, vol. 7, pp. 179–188, 1936.

[5] C. Liu and H. Wechsler, "Evolutionary pursuit and its application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 6, pp. 570–582, Jun. 2000.

[6] R. Lotlikar and R. Kothari, "Fractional-step dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 6, pp. 623–627, Jun. 2000.

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

[8] 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.

[9] P. Navarrete and J. Ruiz-del-Solar, "Kernel-based face recognition by a reformulation of kernel machines," in *Advances in Soft Computing—Engineering, Design, and Manufacturing*, J. Benitez and F. Hoffmann, Eds New York:: Springer Engineering Series, 2003, pp. 183–196.

[10] ——"On the generalization of kernel machines,"——, in *Lecture Notes in Computer Science (Pattern Recognition with Support Vector Machines—SVM 2002)*. New York: Springer, 2002, vol. 2388, pp. 24–39.

[11] ——, "Eigenspace-based recognition of faces: Comparisons and a new approach," in *Proc. 11th Int. Conf. Image Analysis and Processing—ICIAP 2001*, Palermo, Northwest Sicily, Italy, Sep. 2001, pp. 42–47.

[12] ——, "Analysis and comparison of eigenspace-based face recognition approaches," *Int. J. Pattern Recog. Artif. Intell.*, vol. 16, no. 7, pp. 817–830, Nov. 2002.

[13] A. Pentland and B. Moghaddam, "Probabilistic visual learning for object representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 696–710, Jul. 1997.

[14] P. J. Phillips, H. Wechsler, J. Huang, and P. Rauss, "The FERET database and evaluation procedure for face recognition algorithms," *Image Vision Comput. J.*, vol. 16, no. 5, pp. 295–306, 1998.

[15] J. Ruiz-del-Solar and P. Navarrete, "Toward a generalized eigenspace-based face recognition system," in *Lecture Notes in Computer Science (Structural, Syntactic, and Statistical Pattern Recognition)*. New York: Springer, 2002, vol. 2396, pp. 662–671.

[16] S. Santini and R. Jain, "Similarity measures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 9, pp. 871–883, Sep. 1999.

[17] L. Sirovich and M. Kirby, "A low-dimensional procedure for the characterization of human faces," *J. Opt. Soc. Amer. A*, vol. 4, no. 3, pp. 519–524, 1987.

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

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

[20] Yale University Face Image Database [Online]. Available: http://cvc.yale.edu/ projects/yalefaces/yalefaces.html.

Javier Ruiz-del-Solar (S'89-M'98-SM'04) received the diploma in electrical engineering and the M.S. degree in electronic engineering from the Technical University Federico Santa Maria, Chile, in 1991 and 1992, respectively, and the Doctor-Engineer degree from the Technical University of Berlin, Germany, in 1997.

In 1998 he joined the Electrical Engineering Department of the University as an Assistant Professor. In 2001 he became Director of the Electro-Technologies Laboratory, and in 2005 Associate Professor. His research interests include Face Analysis, Application of Computational Intelligence Technologies to Computer Vision and Pattern Recognition Problems, and Mobile Robotics.

Pablo Navarrete (S'01) received the B.S. degree in physics from the University of Chile in 2000 and the electrical engineering diploma from the University of Chile in 2002. He is currently pursuing the Ph.D. degree in electrical engineering at Purdue University, West Lafayette, IN, with major area in communication and signal processing.